192
1/13/81
T-5
R1190
D.2189

ORNL/TM-7369

ornl

OAK RIDGE NATIONAL LABORATORY

UNION CARBIDE

MASTER

System Software for the NMFECC CRAY-1 Version of GIFTS 4B

W. H. Gray
T. V. Baudry

OPERATED BY
UNION CARBIDE CORPORATION
FOR THE UNITED STATES
DEPARTMENT OF ENERGY

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency Thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

DISCLAIMER

Portions of this document may be illegible in electronic image products. Images are produced from the best available original document.

Printed in the United States of America. Available from
National Technical Information Service
U.S. Department of Commerce
5285 Port Royal Road, Springfield, Virginia 22161
NTIS price codes—Printed Copy: A07 Microfiche A01

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

ORNL/TM-7369
Dist. Category UC-32

Contract No. W-7405-eng-26

FUSION ENERGY DIVISION

SYSTEM SOFTWARE FOR THE NMFECC CRAY-1 VERSION OF GIFTS 4B

W. H. Gray

T. V. Baudry

Date Published: January 1981

DISCLAIMER

This book was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

Prepared by the
OAK RIDGE NATIONAL LABORATORY
Oak Ridge, Tennessee 37830
operated by
UNION CARBIDE CORPORATION
for the
DEPARTMENT OF ENERGY

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

THIS PAGE

WAS INTENTIONALLY

LEFT BLANK

CONTENTS

ACKNOWLEDGMENTS

We would like to express our appreciation to W. C. T. Stoddart and W. D. Cain for their suggestions and encouragement for the NMFECC CRAY-1 version of GIFTS and to H. A. Kamel, M. W. McCabe, and W. Spector for their useful conversations about the GIFTS system.

ABSTRACT

The Oak Ridge National Laboratory (ORNL) maintains a version of the GIFTS system structural analysis computer programs. Executable modules are supported on two different types of computer hardware, a DECsystem-1Ø and a CRAY-1. Without external difference to the user, these modules execute equivalently upon both types of hardware. Presented herein are the local software enhancements for the ORNL version of GIFTS for the National Magnetic Fusion Energy Computer Center (NMFECC) CRAY-1 computer as well as a description of the ORNL implementation of the system-dependent portions of the GIFTS software library for the NMFECC CRAY-1.

1. INTRODUCTION

The Oak Ridge National Laboratory (ORNL) maintains a version of the GIFTS[1] system structural analysis computer programs on the National Magnetic Fusion Energy Computer Center (NMFECC) CRAY-1 computer. Except for the increased execution speed and numerical accuracy, the performance of the CRAY-1 GIFTS executable modules is equivalent to that of their counterparts on the ORNL DECsystem-1Ø.[2] This document describes the local enhancements to the GIFTS system available on the NMFECC CRAY-1 version as well as the system-dependent portions of the GIFTS software library that have been written and are maintained for the CRAY-1. This document has the following organization and is not intended for the casual user of GIFTS.

In Sect. 2 the graphics software required by the GIFTS system is presented. The special features of the NMFECC CRAY-1 version of GIFTS discussed in Sect. 2 include (1) using a locally developed graphics package to create an intermediate plot data format file to generate report-quality graphics and (2) creating a standard Tektronix Terminal Control System (TCS) software library for the CRAY-1.

The GIFTS system I/O subprograms are presented in Sect. 3. This software uses FORTLIB[3] to open, create, close, read, and write direct access and sequential disk files. The GIFTS system encoding and decoding software, presented in Sect. 4, uses the extensive character-manipulating capabilities of the BASELIB[4] software library.

The GIFTS system bit packing/unpacking software and timing software are discussed in Sects. 5 and 6, respectively. This software is also written using BASELIB subroutines.

Section 7 presents the added ORNL GIFTS commands and explains how these commands are easily maintained through updates of the GIFTS system. Briefly stated, these new commands allow an expanded graphics capability in the area of report-quality plot generation. In Sect. 8 three miscellaneous supporting subprograms for the NMFECC CRAY-1 version of GIFTS are discussed.

Section 9 presents the TCS software written to create a library of working TCS subroutines for the NMFECC CRAY-1 version of GIFTS. In Sect. 1Ø the software that was written to emulate the TCSTKF software library[5] is presented. This section controls creation of the intermediate device-independent graphics files.

The executable modules are located in FILEM under user number 2304 and directory .GIFTSV4B. Their names are a concatenation of the module name the user wishes to execute followed by the two characters -C. Therefore, to read a copy of the executable module for BULKM and begin execution, the following might be typed:

```
RFILEM RDS 2304 .GIFTSV4B BULKM-C ! END / 1 1

BULKM-C / 5 1
```

Presently, the executable modules are protected so that only divisional read-only access is permitted. The source and library files are protected so that only user 2304 has access to them.

2. NMFECC CRAY-1 GIFTS GRAPHICS SOFTWARE

The LIB5 GIFTS software library makes use of the Tektronix PLOT 1Ø TCS[6,7] software library to display images on the screen of a Tektronix 4ØØØ series storage tube terminal. The NMFECC CRAY-1 version of GIFTS uses the standard level 3.3 version of the Tektronix PLOT 1Ø. For further details about the TCS software library used on the CRAY-1 for the GIFTS plotting programs, see Sect. 9.

The NMFECC CRAY-1 version of GIFTS also uses a modified form of the TCSTKF[5] software library (the name TCSTKF refers to the enhanced TCS software library) to create intermediate .TKF plot data files. These files are automatically created under user control (see Sect. 7 for a complete description of the ORNL GIFTS commands that control creation of the intermediate plot data files and Sect. 1Ø for a presentation of the CRAY-1 version of the TCSTKF software library). This locally written software library performs the function by copying the exact character stream that is used to create an image on the screen of the user's Tektronix terminal into a disk file appropriately named XJOB.TKF [where XJOB represents the four-character job name and TKF represents Tektronix character (K) File]. The creation of the appropriately named .TKF file is coordinated by subroutine URSCMD.

The intermediate .TKF file can be postprocessed in a variety of ways. The DECsystem-1Ø monitor level PLOT command has been locally modified[8] by ORNL to permit access to any graphics device at ORNL. For example, to obtain a display image of the file HOLE.TKF on a system-resident Versatec printer/plotter, the following command can be issued (this example assumes that the file HOLE.TKF has been transmitted back to the ORNL DECsystem-1Ø using the binary file option of NETOUT[9]):

```
.PLOT VER:=HOLE.TKF
```

The following sections present the ORNL GIFTS graphics software currently being executed.

2.1 SUBROUTINE CURSOR

The function of subroutine CURSOR is to force the Tektronix terminal into the graphic input mode (GIN). In the GIN mode the terminal graphic cross-hair cursor is turned on, and the terminal is ready to transmit both a character and the coordinates of the cross-hair cursor to the calling program. Warning: when the terminal is in the GIN mode, do not strike the return key to transmit the coordinates. Doing so will confuse the TCS software while it is trying to decode the cursor coordinates.

Calling sequence:

CALL CURSOR (CH,IX,IY)

Arguments:

CH	is the character struck in response to the graphic cross-hair cursor. CH is a zero-filled, right-justified character.
IX	is the graphic cross-hair cursor's x-coordinate position, returned.
IY	is the graphic cross-hair cursor's y-coordinate position, returned.

Language: CFT-FORTRAN

Subroutine listing:

```
C
        SUBROUTINE CURSOR(ICH,IX,IY)
        CALL DCURSR (ICH,IX,IY)
C
C \/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
C \/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
C \/\                                                               \/\
C \/\ THE ENCLOSED CODE IS DEPENDENT UPON THE COMPUTER SYSTEM BEING \/\
C \/\ USED.  THIS IS THE VERSION TO BE USED ON A CRAY-1 COMPUTER.  \/\
C \/\                                                               \/\
C
C  CONVERT TO UPPER CASE IF NECESSARY
        IF (ICH .GE. 141B .AND. ICH .LE. 172B) ICH = ICH - 32
C \/\                                                               \/\
C \/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
C \/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
        RETURN
        END
```

2.2 SUBROUTINE DINIT

The function of subroutine DINIT is to erase and move the beam invisibly to the lower left-hand corner of the Tektronix screen.

Calling sequence:

```
CALL DINIT
```

Arguments: none

Language: CFT-FORTRAN

Subroutine listing:

```
C
      SUBROUTINE DINIT
C
C  RESET THE TEKTRONIX PLOTTING PACKAGE
      CALL ERASE
      CALL OFFSET (Ø)
      CALL MOVEA (Ø.,Ø.)
      RETURN
      END
```

2.3 SUBROUTINE LINA

The function of subroutine LINA is to move the electron beam of the Tektronix storage tube to a specified coordinate in the virtual window.

Calling sequence:

```
CALL LINA (IX,IY,IBVIS)
```

Arguments:

IX	is the virtual x-coordinate where the beam is to be moved.
IY	is the virtual y-coordinate where the beam is to be moved.
IBVIS	is a beam-visibility switch. If IBVIS is greater than Ø, then the beam is to be bright (visible). If IBVIS is less than or equal to Ø, then the beam is to be dark (invisible).

Language: CFT-FORTRAN

Subroutine listing:

```
C
      SUBROUTINE LINA (IXC,IYC,IBVIS)
C
      COMMON  /WHERE/ XC , YC
C
      XC=FLOAT(IXC)
```

```
        YC=FLOAT(IYC)
C
        GO TO (5,1Ø),IBVIS+1
        STOP 171
C
5       CALL MOVEA (XC,YC)
        RETURN
C
1Ø      CALL DRAWA (XC,YC)
        RETURN
        END
```

2.4 SUBROUTINE LINE

The function of subroutine LINE is to move the electron beam of the Tektronix storage tube to a specified coordinate in the virtual window relative to the previous position.

Calling sequence:

```
CALL LINE (IXR,IYR,IBVIS)
```

Arguments:

IXR — is the virtual x-coordinate where the beam is to be moved relative to the current position.

IYR — is the virtual y-coordinate where the beam is to be moved relative to the current position.

IBVIS — is a beam-visibility switch. If IBVIS is greater than Ø, then the beam is to be bright (visible). If IBVIS is less than or equal to Ø, then the beam is to be dark (invisible).

Language: CFT FORTRAN

Subroutine listing:

```
C
        SUBROUTINE LINE (IXC,IYC,IBVIS)
C
        COMMON  /WHERE/ XC , YC
C
        XC=XC+FLOAT(IXC)
        YC=YC+FLOAT(IYC)
C
        GO TO (5,1Ø),IBVIS+1
        STOP 172
C
5       CALL MOVEA (XC,YC)
```

```
        RETURN
C
1Ø      CALL DRAWA (XC,YC)
        RETURN
        END
```

2.5 SUBROUTINE OFFSET

Setting the virtual and screen window for the Tektronix software is the function of subroutine OFFSET. The GIFTS system divides the plotting area into model statistics and model viewing area. This subprogram prevents the overlapping of the two areas.

Calling sequence:

```
CALL OFFSET (IOFF)
```

Arguments:

IOFF — is the binary switch that indicates whether the model statistics or model viewing area is to be selected. If IOFF is Ø, then the model viewing area is appropriately windowed. If IOFF is 1, then the model statistics area is appropriately windowed.

Language: CFT-FORTRAN

Subroutine listing:

```
C
        SUBROUTINE OFFSET (I)
        GO TO (5,1Ø),I+1
        STOP 173
C
C  WINDOW FOR MODEL PLOTTING
5       CONTINUE
        CALL VWINDO (Ø.,8ØØ.,Ø.,8ØØ.)
        CALL SWINDO (Ø,799,Ø,799)
        RETURN
C
C  WINDOW FOR THE LABEL
1Ø      CALL VWINDO (Ø.,223.,Ø.,8ØØ.)
        CALL SWINDO (799,223,Ø,799)
        RETURN
        END
```

2.6 SUBROUTINE SETPT

The function of subroutine SETPT is to move the electron beam of the Tektronix storage tube invisibly to the specified coordinates in the virtual window.

Calling sequence:

```
CALL SETPT (IXC,IYC)
```

Arguments:

IX is the virtual x-coordinate where the beam is to be invisibly moved.

IY is the virtual y-coordinate where the beam is to be invisibly moved.

Language: CFT-FORTRAN

Subroutine listing:

```
C
      SUBROUTINE SETPT (IXC,IYC)
C
      COMMON  /WHERE/ XC , YC
C
      XC=FLOAT(IXC)
      YC=FLOAT(IYC)
      CALL MOVEA (XC,YC)
      RETURN
      END
```

2.7 SUBROUTINE TBELL

The function of subroutine TBELL is to ring the terminal bell in the Tektronix terminal as an auditory prompt for input. However, if the job is being controlled by something other than the user's terminal, the bell is not rung.

Calling sequence:

```
CALL TBELL
```

Arguments: none

Language: CFT-FORTRAN

Subroutine listing:

```
C
        SUBROUTINE TBELL
C
C \/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
C \/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
C \/\                                                               \/\
C \/\ THE ENCLOSED CODE IS DEPENDENT UPON THE COMPUTER SYSTEM BEING \/\
C \/\ USED.  THIS IS THE VERSION TO BE USED ON A CRAY-1 COMPUTER.   \/\
C \/\                                                               \/\
C
        LOGICAL BATCHJ
C
        DATA IBELCRLF/ 34070160340701606412B/
        IF (BATCHJ (0)) RETURN
C
C  SEND A MESSAGE TO THE CONTROLLER
        IF (IZSNDMR (IBELCRLF,8,0) .NE. 0) 100,100
100     CONTINUE
        RETURN
C \/\                                                               \/\
C \/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
C \/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
        END
```

2.8 SUBROUTINE TERASE

Erasing the screen of the Tektronix storage tube is the function of subroutine TERASE.

Calling sequence:

CALL TERASE

Arguments: none

Language: CFT-FORTRAN

Subroutine listing:

```
C
        SUBROUTINE TERASE
C *** WRITTEN DECEMBER 13, 1977, BY MICHAEL W. MCCABE
C
C     SUBROUTINE TO ERASE THE TEKTRONIX SCREEN.
C
        CALL ERASE
        CALL ANMODE
        CALL TSEND
        RETURN
        END
```

2.9 SUBROUTINE TEXT

The function of subroutine TEXT is to write characters on the screen of a Tektronix terminal at a preselected position. The characters are stored in an xA4 format.

Calling sequence:

CALL TEXT (NUMBER,CHARAC)

Arguments:

NUMBER	is the number of characters to be output to the terminal.
CHARAC	is the array containing the characters (stored with an xA4 format). The alphanumeric terminal cursor should already have been moved to the screen coordinates where the text is to be drawn.

Language: CFT-FORTRAN

Subroutine listing:

```
C
        SUBROUTINE TEXT (T,N)
C
        DIMENSION T(3) , KADE(4)
C
        IO=0
        DO 100 I=1,3
C
C \/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
C \/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
C \/\                                                            \/\
C \/\ THE ENCLOSED CODE IS DEPENDENT UPON THE COMPUTER SYSTEM BEING \/\
C \/\ USED.  THIS IS THE VERSION TO BE USED ON A CRAY-1 COMPUTER.   \/\
C \/\                                                            \/\
C
C  CONVERT FROM A4 ASCII TO ADE CHARACTERS
        CALL ZRJCHARZ (KADE,T(I),4)
C \/\                                                            \/\
C \/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
C \/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
        DO 100 J=1,4
        CALL ANCHO (KADE(J))
        IO=IO+1
        IF (IO .GE. N) GO TO 105
100     CONTINUE
105     CONTINUE
        RETURN
        END
```

2.1Ø SUBROUTINE THOME

Moving the alphanumeric cursor to its home position on the screen of the Tektronix storage tube is the function of subroutine THOME.

Calling sequence:

```
     CALL THOME
```

Arguments: none

Language: CFT-FORTRAN

Subroutine listing:

```
C
        SUBROUTINE THOME
C *** WRITTEN DECEMBER 13, 1977, BY MICHAEL W. MCCABE
C
C     THIS SUBROUTINE CHANGES THE TEKTRONIX TERMINAL TO ALPHA
C     MODE WITH THE CURSOR POSITIONED IN THE HOME POSITION.
C
        CALL HOME
        CALL ANMODE
        CALL TSEND
        RETURN
        END
```

2.11 SUBROUTINE TINITT

The function of subroutine TINITT is to initialize the Tektronix PLOT 1Ø TCS software library.[6] For the NMFECC CRAY-1 version of GIFTS, a call is made to subroutine LINK (if necessary) to create a drop file and connect the controller to the user's terminal (FORTRAN logical unit number 59).

Calling sequence:

```
     CALL TINITT
```

Arguments: none

Language: CFT-FORTRAN

Subroutine listing:

```
C
        SUBROUTINE TINITT
C *** UPDATED: 31JAN79 BY WHG
C *** UPDATED 12/19/77 BY MWM
C *** WRITTEN DECEMBER 13, 1977, BY MICHAEL W. MCCABE
C
C     THIS ROUTINE INITIALIZES THE PLOTTING PACKAGE AND TELLS
```

```
C      IT THE BAUD RATE AT WHICH TO TRANSMIT TO THE TERMINAL.
C
C
C \/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
C \/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
C \/\                                                                 \/\
C \/\ THE ENCLOSED CODE IS DEPENDENT UPON THE COMPUTER SYSTEM BEING \/\
C \/\ USED.  THIS IS THE VERSION TO BE USED ON A CRAY-1 COMPUTER.   \/\
C \/\                                                                 \/\
      LOGICAL LINKCALL
C
      COMMON /INITDROP/ LINKCALL
      COMMON /JOB/ XJOB
C
C
C  TELL LINK UNIT59 IS THE TERMINAL AND CREATE A DROP FILE, IF NECESSARY
      IF (.NOT. LINKCALL) CALL LINK ("UNIT59=TERMINAL//")
      LINKCALL = .TRUE.
C \/\                                                                 \/\
C \/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
C \/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
C
C  INITIALIZE THE TEKTRONIX SOFTWARE
      CALL INITT (120)
      CALL ANMODE
      RETURN
      END
```

3. NMFECC CRAY-1 UDB I/O SOFTWARE

The GIFTS system manipulates a set of files in the user's disk area collectively called a unified data base (UDB). Within the UDB all the necessary information about a job is stored in a consistent manner so that individual GIFTS modules can perform functions based on these data using instructions from the user. Most of the UDB files for the CRAY-1 version of GIFTS use the FORTRAN direct access I/O mode.

Subroutine DEFIN creates these direct access files, each of which has a specific physical record size (the number of words contained on one physical record) and number of records (the length of the file). GIFTS usually defines multiple logical records within one physical record; however, at this primitive level of software coding, knowledge of the number of logical records per physical record is irrelevant. The length of the direct access file is used to ensure that the file is always preset to Ø.

The GIFTS software library must also provide for sequential file definition and access. The subroutines OPENIF and OPENOF allow for this type of file handling. Due to the present system architecture of the NMFECC CRAY-1, public disk areas are not available for programs such as these that have relatively few users. Therefore, both the GIFTS help file and the estimate file must be in the user's disk area if either is to be accessed. Both the files are located in FILEM under user number 2304 directory .GIFTSV4B. Divisional read-only access is permitted.

Finally, subroutine SLTASN is used to assign FORTRAN logical unit numbers for use by the GIFTS system during access to the UDB. There can be a maximum of five files open simultaneously (a GIFTS limitation). The FORTRAN logical unit numbers chosen are 16 through 2Ø, inclusive.

Also described in the following sections are subroutines that read from, write to, rename, delete, and close direct access UDB files.

3.1 SUBROUTINE CLOSEF

The function of subroutine CLOSEF is to close a file, if one is active, on a given FORTRAN logical unit number. Because of the CRAY-1 operating system, special handling of the GIFTS estimate file must be performed. As described in subroutine OPENOF, whenever the file GIFT.EST is opened for sequential output, the file GIFT.ES is really opened to avoid the conflict of having two files with the same name open simultaneously on different FORTRAN logical unit numbers. Therefore, every time the file GIFT.EST is closed, it is deleted, and every time the file GIFT.ES is closed, it is renamed to GIFT.EST.

Calling sequence:

CALL CLOSEF (ISL)

Arguments:

ISL	is the FORTRAN logical unit number associated with the file to be closed.

Language: CFT-FORTRAN

Subroutine listing:

```
C
      SUBROUTINE CLOSEF (ISL)
C
C \/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
C \/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
C \/\                                                               \/\
C \/\ THE ENCLOSED CODE IS DEPENDENT UPON THE COMPUTER SYSTEM BEING \/\
C \/\ USED.  THIS IS THE VERSION TO BE USED ON A CRAY-1 COMPUTER.   \/\
C \/\                                                               \/\
C
      LOGICAL   PRESNT
C
      COMMON /FILBUFIO/     ISLISR(5,2)
      COMMON /JOB/   XJOB , XJOB2
C
      DATA    GIFT/   4HGIFT/,  EST/    4H EST/
C
      CALL CLOSE (ISL)
C
C  IF THE FILE WAS GIFT.ES, THEN DELETE THE OLD GIFT.EST FILE
C  AND RENAME THE NEW ONE
      J=ISL-15
      IF (ISLISR(J,1) .NE. 4H ES ) GO TO 100
      SAVN=XJOB
      XJOB=GIFT
```

```
C
C  DELETE THE OLD OLD, IF PRESENT
        IF (PRESNT(EST)) CALL DESTROY (8HGIFT.EST)
        XJOB=SAVN
C
C  NOW RENAME THE NEW OLD
        CALL CHGNAME (8HGIFT.ES ,8HGIFT.EST)
1ØØ     CONTINUE
        ISLISR(J,1)=Ø
        ISLISR(J,2)=Ø
C \/\                                                                    \/\
C \/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
C \/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
        RETURN
        END
```

3.2 SUBROUTINE DEFIN

Opening, expanding, or creating a direct access file is the function of subroutine DEFIN. If the file is expanded or created, then the newly created portion of the file will be set to Ø. The NMFECC CRAY-1 operating system does not allow a file to be expanded; therefore it is necessary to create a temporary file of the newly specified larger size, copy the contents of the old file into the new file, delete the old file, zero out the rest of the new file, and finally rename the temporary file to the old file name.

Calling sequence:

```
CALL DEFIN (ISL,ISR,INR,FNAME,IREDEF,ISZERO)
```

Arguments:

ISL	is the FORTRAN logical unit number associated with the file to be opened.
ISR	is the physical record size of the file in words.
INR	is the total number of physical records to be defined for the file.
FNAME	is the file name, an ASCII text string stored in 2A5 format.
IREDEF	is a binary switch that indicates redefinition (expansion) of the file. If IREDEF is equal to Ø, then the file may have been previously opened. If IREDEF is equal to 1,

then the file will be reopened to a larger size.

ISZERO is the first record number to be set to Ø (valid only during file creation or redefinition).

Language: CFT-FORTRAN

Subroutine listing:

```
      SUBROUTINE DEFIN (ISL,ISR,INR,NAME,IREDEF,ISZERO)
C UPDATED: 21JAN79 WHG ***
C
C \/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
C \/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
C \/\                                                                        \/\
C \/\ THE ENCLOSED CODE IS DEPENDENT UPON THE COMPUTER SYSTEM BEING           \/\
C \/\ USED.  THIS IS THE VERSION TO BE USED ON A CRAY-1 COMPUTER.            \/\
C \/\                                                                        \/\
C *** UPDATED 11/25/77 BY MWM
C *** WRITTEN DECEMBER 1Ø, 1975, BY MICHAEL W. MCCABE
C
C     THIS SUBROUTINE OPENS A DIRECT ACCESS FILE, USING THE
C     FOLLOWING PARAMETERS:
C
C       ISL    - LOGICAL UNIT NUMBER TO WHICH TO ATTACH THE FILE
C       ISR    - SIZE OF EACH RECORD IN FILE (IN WORDS)
C       INR    - NUMBER OF RECORDS IN FILE
C       NAME   - ARRAY CONTAINING NAME OF FILE TO BE OPENED
C       IREDEF - REDEFINITION INDICATOR:
C                  Ø - NO REDEFINITION
C                  1 - REDEFINITION OF EXISTING FILE
C       ISZERO - ZEROIZATION INDICATOR:
C                  Ø - NO ACTION
C                 Ø - ZEROIZE RECORDS 'ISZERO' THROUGH 'INR'
C                      (VALID ONLY DURING CREATION OR REDEFINITION)
C
      COMMON  /CMDSLT/        ISLIN , ISLOUT , ISLTTY
C ------------------- UPDATE #6 --------------------
      COMMON /FILBUF/FB,1B(5)
C ----------------- MWM  11/25/77 ------------------
      COMMON /FILBUFIO/       ISLISR(5,2)
C
      LOGICAL PRESNT
C
      DIMENSION       NAME(2) , IBUFF(1ØØØ)
C
      DATA    IBUFF/  1ØØØ*Ø/ , IFTMP/    4H ZZZ/
C
C  FIRST STORE PHYSICAL RECORD SIZE INTO FILBUFIO
      J=ISL-15
      ISLISR(J,1)=NAME(2)
      ISLISR(J,2)=ISR
C
```

```
C  MAKE THE PACKED FILE NAME
        CALL PERMN (NAME,FLNM)
C
C  DEFINE THE TYPE (DIRECT ACCESS) AND THE LENGTH OF THE FILE
        ITYPE=4
        LEN=ISR*INR
C
C  IS THIS A REDEFINITION
        IF (IREDEF .GT. Ø) GO TO 1Ø
C
C  NO IT IS NOT, HAS THE FILE BEEN CREATED YET.
        IF ( .NOT. PRESNT(NAME(2))) GO TO 2Ø
C
C  YES IT IS CREATED, SIMPLY OPEN IT.
        CALL OPEN (ISL,FLNM,ITYPE,LEN)
        RETURN
2Ø      CONTINUE
C
C  NO IT HAS NOT, THEREFORE CREATE IT.
        CALL CREATE (ISL,FLNM,ITYPE,LEN)
C
C  NOW ZERO OUT THE FILE
C
C  FIRST CHECK FOR IBUFF EXCEEDED ( THIS IS A SYSTEM ERROR - MINE)
        IF (ISR .GT. 1ØØØ) STOP 6236
C
        DO 3Ø I=1,ISR
        IBUFF(I)=Ø
3Ø      CONTINUE
C
C  LOOP THROUGH ALL THE RECORDS
        DO 35 I=1,INR
        CALL OUTRA (ISL,I,IBUFF,ISR)
35      CONTINUE
        RETURN
1Ø      CONTINUE
C
C  MAKE A TEMPORARY FILE FOR THE REDEFINITION
        ISAVN=NAME(2)
        NAME(2)=IFTMP
        CALL PERMN (NAME,FTLNM)
        NAME(2)=ISAVN
        CALL CREATE (ISL,FTLNM,ITYPE,LEN)
C
C  FIND A FREE UNIT FOR THE COPY OPERATION
        CALL FREEUS (ISLOT)
C
C  OPEN THE OLD FILE
        IM=ISZERO-1
        LENO=ISR*IM
        CALL OPEN (ISLOT,FLNM,ITYPE,LENO)
C
```

```
C  NOW TRANSFER THE PREVIOUSLY DEFINED RECORDS
         ASSIGN 1ØØ TO IERROR
         DO 4Ø I=1,IM
         CALL RDABS (ISLOT,IBUFF,ISR,(I-1)*ISR,IERROR)
         CALL OUTRA (ISL,I,IBUFF,ISR)
4Ø       CONTINUE
C
C  ZEROIZE THE NEW RECORDS
         DO 45 I=1,ISR
         IBUFF(I)=Ø
45       CONTINUE
         DO 5Ø I=ISZERO,INR
         CALL OUTRA (ISL,I,IBUFF,ISR)
5Ø       CONTINUE
C
C  BOOKKEEP FILE NAMES
         CALL CLOSE (ISL)
         CALL CLOSE (ISLOT)
         CALL DESTROY (FLNM)
         CALL CHGNAME (FTLNM,FLNM)
         CALL OPEN (ISL,FLNM,ITYPE,LEN)
         RETURN
C
C  ERROR DURING RDABS BRANCH
1ØØ      CONTINUE
         WRITE(ISLTTY,1ØØØ) FLNM
1ØØØ     FORMAT('?DEFIN/GIFTS: READ ERROR IN FILE ',A8,
     1   ' DURING REDEFINITION')
C \/\                                                                    \/\
C \/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
C \/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
         STOP 6237
         END
```

3.3 SUBROUTINE DELETE

The function of subroutine DELETE is to delete a specified file from the user's disk area and to update the XJOB.FIL file accordingly.

Calling sequence:

```
CALL DELETE (XEXT)
```

Arguments:

XEXT is the extension of the file to be deleted.

Language: CFT-FORTRAN

Subroutine listing:

```
C
      SUBROUTINE DELETE (EXT)
C *** UPDATED: 21JAN79 WHG ***
C *** LAST UPDATED 11/25/77 BY MWM
C *** WRITTEN JANUARY 6, 1976, BY MICHAEL W. MCCABE
C
C     THIS SUBROUTINE DELETES FILE JOB,EXT FROM DISK AND ALSO
C     DELETES ITS SPECIFICATIONS FROM FILE JOB,FIL.
C
      LOGICAL FILDEF
C
      DIMENSION LFSZ(5)
C
      COMMON /JOB/XJOB,XJOB2
C -------------------- UPDATE #6 --------------------
      COMMON /PAR/NWF
C ------------------ MWM  11/25/77 ------------------
C
      DATA BLANK/Ø./
C
C *** FIND FREE SLOT.
      CALL SLTASN (ISL)
C
C *** DELETE THE FILE FROM DISK.
      XJOB2=EXT
C
      CALL PERMN (XJOB,FLNM)
C
C \/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
C \/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
C \/\                                                                          \/\
C \/\ THE ENCLOSED CODE IS DEPENDENT UPON THE COMPUTER SYSTEM BEING             \/\
C \/\ USED.  THIS IS THE VERSION TO BE USED ON A CRAY-1 COMPUTER.               \/\
C \/\                                                                          \/\
      CALL DESTROY (FLNM)
C \/\                                                                          \/\
C \/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
C \/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
C
C *** DELETE ITS SPECIFICATIONS FROM JOB,FIL.
      IF (FILDEF(ISL,EXT,I,LFSZ)) CALL OUTRA (ISL,I,BLANK,NWF)
C ------------------ MWM  11/25/77 ------------------
C
C *** RELEASE SLOT 'ISL'.
      CALL CLOSEF (ISL)
      CALL FRESLT (ISL)
      RETURN
      END
```

3.4 SUBROUTINE FREAD

Performance of free-formatted sequential I/O is the function of subroutine FREAD. For the ORNL version of GIFTS, this subprogram has been modified to allow (1) tabs (ASCII character 11_8) to be interpreted as spaces and (2) the dollar sign, the exclamation point, and the semicolon (ASCII characters 44_8, 41_8, and 73_8, respectively) to be interpreted as the beginning of inter-record comments. Because of the NMFECC CRAY-1 operating system use of lower case ASCII, all input upper-case alphabetic characters are converted to lower case.

Calling sequence:

```
CALL FREAD (IT,N,L,V)
```

Arguments:

IT	is an integer switch used to determine what type of I/O is to be done. If IT is equal to 1, then integer values are to be read. If IT is equal to 2, then floating-point values are to be read. If IT is equal to 3, then alphanumeric values are to be read. If IT is equal to 4, then one alphanumeric and one integer are to be read.
N	is the number of items to be read.
L	is an array for the return of the integer values read by this subprogram.
V	is an array for the return of the floating-point values read by this subprogram.

Language: CFT-FORTRAN

Subroutine listing:

```
      SUBROUTINE FREAD (IT,N,L,V)
C UPDATED: 21JAN79 WHG ***
C *** UPDATED 9/26/76 BY WHG FOR CRAY-1 CHARACTERS
C *** WRITTEN FEBRUARY 3, 1976 BY HUSSEIN A. KAMEL
C
C ***************************************************************
C ***                                                         ***
C ***                    FREE FORMATTED READ                  ***
C ***                                                         ***
C ***          IT = 1 -- INTEGER                              ***
C ***             = 2 -- FLOATING POINT                       ***
```

```
C ***              = 3 -- ALPHANUMERIC                                  ***
C ***              = 4 -- ONE ALPHA. + ONE INTEGER                      ***
C ***                                                                   ***
C ***          N = NO. OF ITEMS TO BE READ                              ***
C ***          L = ARRAY FOR RETURN OF INTEGERS                         ***
C ***          V = ARRAY FOR RETURN OF ALPHA + FLOATING POINT           ***
C ***                                                                   ***
C *************************************************************************
C
      INTEGER BLANK,COMMA,ZERO,E,PLUS,PERIOD,SEMI,EXCLM,TAB,DOLLAR
C
      LOGICAL COMNT
C
      DIMENSION L(1),V(1),IB(81),K(2),KA(8)
      COMMON /CMDSLT/ISLIN,ISLOUT,ISLTTY
C
      DATA BL,CMA/1H ,1H,/
C
C \/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
C \/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
C \/\                                                                   \/\
C \/\ THE ENCLOSED CODE IS DEPENDENT UPON THE COMPUTER SYSTEM BEING      \/\
C \/\ USED.  THIS IS THE VERSION TO BE USED ON A CRAY-1 COMPUTER.        \/\
C \/\                                                                   \/\
      DATA IB(81)/32/
       DATA  BLANK/32/,  COMMA/44/,  MINUS/45/,  PLUS  /43/
       DATA  E    /145B/,  ZERO /48/,  NINE /57/,  PERIOD/46/
      DATA  SEMI/73B/,  EXCLM/33/,  TAB/9/,  DOLLAR/36/
C \/\                                                                   \/\
C \/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
C \/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
C
C \/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
C /\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/
C   THE FOLLOWING FUNCTION IS NECESSARY ONLY FOR THE CRAY-1          /\/\
      KCH(K)=K
C
C /\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/
C \/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
C
C
C ***********************************************
C *** CHARACTER TYPE, NO. OF ITEMS TO BE READ ***
C ***********************************************
C
999   ISTOP=0
998   ICT=IT
      NIT=N
      GO TO (991,992,993,994),IT
C *** INTEGER MODE
991   DO 980 I=1,NIT
980   L(I)=0
```

```
        GO TO 1
C *** REAL VARIABLE MODE
992     DO 981 I=1,NIT
981     V(I)=0.
        GO TO 1
C *** ALPHANUMERIC MODE
993     DO 982 I=1,NIT
        II=2*I
        V(II-1)=BL
982     V(II)=BL
        GO TO 1
C *** 1 ALPHANUMERIC + 1 INTEGER
994     V(1)=BL
        V(2)=BL
        L(1)=0
        ICT=3
        NIT=2
C
C *****************************
C *** INITIALIZE PARAMETERS ***
C *****************************
C
C *** INITIALIZE NO. OF ITEMS
1       IF (ISTOP .EQ. 1) RETURN
        NI=1
C
C *******************
C *** READ BUFFER ***
C *******************
C
        READ (ISLIN,1000) (IB(I),I=1,80)
1000    FORMAT (80R1)
C
C *** ECHO LINE IF STEERING FILE COMMAND.
        IF (IT .NE. 4 .OR. ISLIN .EQ. ISLTTY) GO TO 20
        DO 5 I=1.80
        J=81-I
        IF (KCH(IB(J)) .NE. BLANK) GO TO 10
5       CONTINUE
10      WRITE (ISLTTY,15) (IB(I),I=1,J)
15      FORMAT (1X,80R1)
C
C ********************************************
C *** INITIALIZE START AND END OF ITEM   *****
C ********************************************
C
20      CONTINUE
C
C \/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
C \/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
C \/\                                                               \/\
C \/\ THE ENCLOSED CODE IS DEPENDENT UPON THE COMPUTER SYSTEM BEING \/\
```

```
C \/\ USED.  THIS IS THE VERSION TO BE USED ON A CRAY-1 COMPUTER.   \/\
C \/\                                                              \/\
C
C ... BLANK OUT COMMENTS BEFORE PASSING TO DECODERS
C ... THE FOLLOWING SYMBOLS ARE COMMENTS
C ...   $       ;
C
C ... ALSO CONVERT TABS TO SPACES
        COMNT=.FALSE.
        IF(IB(1) .EQ. SEMI) GO TO 1
        IF(IB(1) .EQ. EXCLM) GO TO 1
        IF(IB(1) .EQ. DOLLAR) GO TO 1
        IF(IB(1) .EQ. TAB) IB(1)=BLANK
        DO 2 I=2,8Ø
        IF(IB(I) .EQ. TAB) IB(I)=BLANK
        IF(IB(I) .EQ. SEMI) COMNT=.TRUE.
        IF(IB(I) .EQ. EXCLM) COMNT=.TRUE.
        IF(IB(I) .EQ. DOLLAR) COMNT=.TRUE.
        IF(COMNT) IB(I)=BLANK
2       CONTINUE
C
C  CONVERT TO LOWER CASE
        DO 3 I=1,8Ø
        IF (IB(I) .GE. 1Ø1B .AND. IB(I) .LE. 132B) IB(I)=IB(I)+32
3       CONTINUE
C \/\                                                              \/\
C \/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
C \/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
        I1=Ø
        I2=1
C *** SET SWITCH TO INDICATE SUCCESS OF FINDING CHARACTER STRING
C
91Ø     IW=Ø
C
C *** EXAMINE CHARACTER
911     K2=KCH(IB(I2))
        IF (K2 .EQ. COMMA) GO TO 912
        IF (K2 .EQ. BLANK) GO TO 913
        IF (IW .EQ. Ø) I1=I2
C *** CHARACTER FOUND
        IW=1
C *** CONTINUE SEARCH FOR SEPARATOR
914     I2=I2+1
C *** TEST IF END OF BUFFER
        IF (I2 .LE. 8Ø) GO TO 911
C ------------------- UPDATE #7 -------------------
        IF (IW .EQ. Ø) RETURN
C ------------------- MWM 3/29/78 -------------------
        GO TO 912
C *** CHECK IF CHARACTER PREVIOUSLY ENCOUNTERED
913     IF (IW .EQ. Ø) GO TO 914
C *** SEARCH ENDS
```

```
912     I2=I2-1
C *** TEST IF ITEM ZERO OR BLANK
        IF (IW .GT. Ø) GO TO 95Ø
C ************************
C *** ADVANCE ITEM  *****
C ************************
C
94Ø     IF (NI .GE. NIT) RETURN
        NI=NI+1
C
C *** SET SWITCH TO INDICATE SUCCESSFUL LOCATION OF SEPARATOR
        IS=Ø
C
C *** COMMENCE SEARCH FOR SEPARATOR
        I1=I2+1
C
C *** CHECK CHARACTER
941     K2=KCH(IB(I1))
        IF (K2 .EQ. BLANK) GO TO 942
        IF (K2 .EQ. COMMA) GO TO 943
        I2=I1
C
C *** SEARCH ENDS
        I1=I1-1
        GO TO 91Ø
C ***  ADVANCE CHARACTER
942     I1=I1+1
C *** TEST FOR E.O.BUFFER
        IF (I1 .GT. 8Ø) GO TO 94Ø
        GO TO 941
C *** COMMA.  CHECK FOR PREVIOUS SEPARATORS
943     IF (IS .EQ. Ø) GO TO 944
        I2=I1-1
        GO TO 94Ø
C
C *** FIRST SEPARATOR
944     IS=1
        GO TO 942
C
C ***************************
C *** INITIALIZE NEW ITEM ***
C ***************************
C
95Ø     NCH=Ø
        IS=Ø
        IM=Ø
        IE=Ø
        VE=Ø
        VM=Ø.
        ISG=1
        IK=Ø
        NXT=I1
```

```
        IEHIT=Ø
        ISIGN=Ø
C *** CHECK VARIABLE TYPE
5ØØ     GO TO (6ØØ,7ØØ,8ØØ),ICT
C
C *********************
C *** INTEGER INPUT ***
C *********************
C
6ØØ     K2=KCH(IB(NXT))
        IF (K2 .LT. ZERO .OR. K2 .GT. NINE) GO TO 6Ø1
C
C *** NUMERAL -- CHECK IF CAPACITY EXCEEDED
        IF (IE .LT. 1ØØØØ) GO TO 6Ø3
        ISTOP=1
        IF (ISLIN .NE. ISLTTY) GO TO 998
        WRITE (ISLTTY,6Ø5)
        GO TO 999
6Ø3     IE=1Ø*IE+K2-ZERO
6Ø6     NXT=NXT+1
        IF (NXT .GT. I2) GO TO 41Ø
        IF (IK .GT. Ø .AND. IM .EQ. Ø) IK=IK+1
        GO TO 6ØØ
C *** CHECK IF 'E'
6Ø1     IF (K2 .NE. E) GO TO 6Ø2
C *** ILLEGAL CHARACTER
1313    ISTOP=1
C
C ... IF NXT IS 81 THEN ASSUME NULL INPUT FOR REST OF VALUES
        IF(NXT.EQ.81) RETURN
        IF (ISLIN .NE. ISLTTY) GO TO 998
        WRITE (ISLTTY,1314) NXT
1314    FORMAT (1ØH CHARACTER,I4,3X,22H ILLEGAL - RETYPE LINE)
        GO TO 999
6Ø5     FORMAT (3ØH ILLEGAL INTEGER - RETYPE LINE)
C *** TEST FOR '-'
6Ø2     IF (K2 .NE. MINUS) GO TO 6Ø4
        IF (ISIGN .GT. Ø) GO TO 1313
        ISIGN=1
        ISG=-1
        GO TO 6Ø6
C *** TEST FOR '+'
6Ø4     IF (K2 .NE. PLUS) GO TO 1313
        IF (ISIGN .GT. Ø) GO TO 1313
        ISIGN=1
        GO TO 6Ø6
C
C *** FINALIZE INTEGER
41Ø     IF (IM .NE. Ø) GO TO 411
        IF (ISG .LT. Ø) IE=-IE
        L(NI)=IE
        IF (IT .EQ. 4) L(1)=L(NI)
```

```
        GO TO 94Ø
411     IF (IK .GT. Ø) IE=IE+1-IK
        L(NI)=IM* 1Ø**(ISG*IE)
        GO TO 94Ø
C
C ***************************************
C *** REAL VARIABLE, CHECK IF NUMERAL ***
C ***************************************
C
7ØØ     K2=KCH(IB(NXT))
        IF (K2 .LT. ZERO .OR. K2 .GT. NINE) GO TO 7Ø1
C *** NUMERAL
        VE=1Ø.*VE+FLOAT(K2-ZERO)
        IF (IK .GT. Ø ) IK=IK+1
7Ø4     NXT=NXT+1
        IF (NXT .GT. I2) GO TO 42Ø
        GO TO 7ØØ
C *** TEST IF 'E'
7Ø1     CONTINUE
        IF (K2 .NE. E) GO TO 7Ø2
C *** CHECK IF 'E' IS FIRST CHARACTER IN NEW WORD
        IF (I1 .EQ. NXT) GO TO 1313
C *** CHECK IF 'E' ALREADY ENCOUNTERED
        IF (VM .NE. Ø.) GO TO 1313
C *** SAVE MANTISSA, INITIALIZE EXPONENT
        VM=VE
        IF (IK .CT. Ø) VM=VM*1Ø.**(1 IK)
        IF (ISG .LT. Ø) VM=-VM
        VE-Ø.
        ISG=1
        IK=Ø
        IEHIT=1
        ISIGN=Ø
        GO TO 7Ø4
C *** CHECK IF '-'
7Ø2     IF (K2 .NE. MINUS) GO TO 7Ø3
        IF (ISIGN .GT. Ø) GO TO 1313
        ISIGN=1
        ISG=-1
        GO TO 7Ø4
C *** CHECK IF '+'
7Ø3     IF (K2 .NE. PLUS) GO TO 7Ø5
        IF (ISIGN .GT. Ø) GO TO 1313
        ISIGN=1
        GO TO 7Ø4
C *** CHECK IF '.'
7Ø5     IF (K2 .NE. PERIOD) GO TO 1313
C *** CHECK IF IN EXPONENT
        IF (VM .NE. Ø.) GO TO 1313
C *** INITIALIZE IK, TO KEEP TRACK OF COMMA
        IF (IK .GT. Ø) GO TO 1313
        IK=1
```

```
        GO TO 7Ø4
C *** FINALIZE REAL VARIABLE
42Ø     IF (VM .NE. Ø.) GO TO 421
        IF (IK .GT. 1) VE=VE/1Ø.**(IK-1)
        IF (ISG .LT. Ø) VE=-VE
        IF (IEHIT .EQ. 1) VE=Ø.Ø
        V(NI)=VE
        GO TO 94Ø
421     CONTINUE
        IF (ISG .LT. Ø) VE=-VE
        V(NI)=VM* 1Ø**VE
        GO TO 94Ø
C
C *************************
C *** ALPHANUMERIC ITEM ***
C *************************
C
8ØØ     NCH=NCH+1
8Ø3     K2=KCH(IB(NXT))
        KA(NCH)=K2
C *** CHECK IF MAXIMUM NO. OF CHARACTERS HAS BEEN EXCEEDED
        IF (NCH .LE. 8) GO TO 8Ø2
        ISTOP=1
        IF (ISLIN .NE. ISLTTY) GO TO 998
        WRITE (ISLTTY,8Ø1)
        GO TO 999
8Ø2     NXT=NXT+1
        IF (NXT .LE. I2) GO TO 8ØØ
8Ø1     FORMAT (34H TOO MANY CHARACTERS - RETYPE LINE)
C
C *** FINALIZE ALPHANUMERIC VARIABLE
C     V(NI),V(NI+1)='NCH' CHARACTERS (PACKED), BLANK FILL, NI+1
43Ø     CONTINUE
        IF (IT .EQ. 4) ICT=1
        NN=NCH
        II=2*NI
        IF (NCH .GT. 4) NN=4
        CALL PAKAL (NN,KA,V(II-1))
        IF (NCH .GT. 4) GO TO 432
        V(II)=BL
        GO TO 94Ø
432     NN=NCH-4
        CALL PAKAL (NN,KA(5),V(II))
        GO TO 94Ø
        END
```

3.5 SUBROUTINE FRESLT

The function of subroutine FRESLT is to release a FORTRAN logical unit number for further use and to set to 0 the corresponding entry in the INUSE array.

Calling sequence:

CALL FRESLT (ISL)

Arguments:

ISL	is the FORTRAN logical unit to be released.

Language: CFT-FORTRAN

Subroutine listing:

```
C
        SUBROUTINE FRESLT (ISL)
C   ***   LAST UPDATED: 27SEP76 WHG   ***
C *** LAST UPDATED 2/22/76 BY MWM
C *** WRITTEN DECEMBER 11, 1975, BY MICHAEL W. MCCABE
C
C     THIS SUBROUTINE FREES THE FILE SLOT ASSOCIATED WITH LOGICAL
C     UNIT NUMBER 'ISL', BY ZEROIZING THE CORRESPONDING ELEMENT
C     OF ARRAY 'INUSE'.
C
        COMMON /DISKIO/INUSE(5)
C
C *** DETERMINE CORRECT ELEMENT NUMBER (J).
C
C
C \/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
C \/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
C \/\                                                               \/\
C \/\ THE ENCLOSED CODE IS DEPENDENT UPON THE COMPUTER SYSTEM BEING \/\
C \/\ USED.  THIS IS THE VERSION TO BE USED ON A CRAY-1 COMPUTER.   \/\
C \/\                                                               \/\
        J=ISL-15
C \/\                                                               \/\
C \/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
C \/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
C
C *** ZEROIZE /INUSE/ INDICATOR.
        INUSE(J)=0
        RETURN
        END
```

3.6 SUBROUTINE INIHVL

Initialization of the hardware value list is the function of subroutine INIHVL.

Calling sequence:

```
     CALL INIHVL
```

Arguments: none

Language: CFT-FORTRAN

Subroutine listing:

```
C
C ---------------------NEW ROUTINE (LIBRARY) --------------------
        SUBROUTINE INIHVL
C *** WRITTEN MARCH 13, 1978, BY MICHAEL W. MCCABE
C
C     THIS SUBROUTINE INITIALIZES THE HARDWARE VALUE LIST FOR THE
C     COMPUTER HARDWARE BEING USED.
C
        COMMON /PAR/NWF1,NWF2,NWI,NRUXM,NRUYM,NRUXO,NRUYO
     1           ,NRUXSC,NRUYSC,NRUXLC,NRUYLC
C
C
C *** SET STORAGE REQUIREMENTS FOR VARIABLES.
C
C \/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
C \/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
C \/\                                                                  \/\
C \/\ THE ENCLOSED CODE IS DEPENDENT UPON THE COMPUTER SYSTEM BEING \/\
C \/\ USED.  THIS IS THE VERSION TO BE USED ON A CRAY-1 COMPUTER.   \/\
C \/\                                                                  \/\
        NWF1=1
        NWF2=2
        NWI=1
C \/\                                                                  \/\
C \/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
C \/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
C
C *** SET DISPLAY SCREEN DIMENSIONS.
        NRUXM=8ØØ
        NRUYM=8ØØ
        NRUXO=223
        NRUYO=8ØØ
C
C *** SET DISPLAY TERMINAL CHARACTER SIZES.
        NRUXSC=1Ø
        NRUYSC=14
        NRUXLC=1Ø
        NRUYLC=14
        RETURN
        END
```

3.7 SUBROUTINE INITIO

Initialization of the terminal I/O characteristics and logical unit in-use indicators is the function of subroutine INITIO. For the NMFECC CRAY-1 version of GIFTS, a call to subroutine LINK is made (if necessary) to create a drop file and connect the controller to the user's terminal (FORTRAN logical unit number 59).

Calling sequence:

```
     CALL INITIO
```

Arguments: none

Language: CFT-FORTRAN

Subroutine listing:

```
C
C
        SUBROUTINE INITIO
C UPDATED: 21JAN79 WHG ***
C *** LAST UPDATED 11/22/77 BY MWM
C *** WRITTEN JANUARY 26, 1976, BY MICHAEL W. MCCABE
C
C     THIS SUBROUTINE INITIALIZES THE LOGICAL UNIT IN-USE INDICATORS
C     AND THE CURRENT 'GIFTS' VERSION NUMBER (GFTVER).
C
C
C \/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
C \/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
C \/\                                                                          \/\
C \/\ THE ENCLOSED CODE IS DEPENDENT UPON THE COMPUTER SYSTEM BEING             \/\
C \/\ USED.  THIS IS THE VERSION TO BE USED ON A CRAY-1 COMPUTER.               \/\
C \/\                                                                          \/\
        LOGICAL LINKCALL
C \/\                                                                          \/\
C \/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
C \/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
C
        COMMON /CMDSLT/ISLIN,ISLOUT,ISLTTY
        COMMON /DISKIO/INUSE(5)
C
C \/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
C \/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
C \/\                                                                          \/\
C \/\ THE ENCLOSED CODE IS DEPENDENT UPON THE COMPUTER SYSTEM BEING             \/\
C \/\ USED.  THIS IS THE VERSION TO BE USED ON A CRAY-1 COMPUTER.               \/\
C \/\                                                                          \/\
        COMMON /INITDROP/ LINKCALL
        COMMON /MACHDPN/NMBITSW,NRIGHTOF
C \/\                                                                          \/\
```

```
C \/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
C \/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
C ------------------- UPDATE #6 -------------------
      COMMON /VERSN/UDBVER,GFTVER
C
      DATA V4B/3HV4B/
C ----------------- MWM  11/22/77 -----------------
C
C
C \/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
C \/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
C \/\                                                                        \/\
C \/\ THE ENCLOSED CODE IS DEPENDENT UPON THE COMPUTER SYSTEM BEING           \/\
C \/\ USED.  THIS IS THE VERSION TO BE USED ON A CRAY-1 COMPUTER.            \/\
C \/\                                                                        \/\
      DATA    LINKCALL/.FALSE./
C
C  TELL LINK UNIT59 IS THE TERMINAL AND CREATE A DROP FILE, IF NECESSARY
      IF (.NOT. LINKCALL) CALL LINK ("UNIT59=TERMINAL//")
      LINKCALL = .TRUE.
C \/\                                                                        \/\
C \/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
C \/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
C
C *** ZEROIZE IN-USE INDICATORS.
      DO 5 I=1,5
5     INUSE(I)=Ø
C
C *** INITIALIZE TELETYPE INPUT AND OUTPUT LOGICAL UNIT NUMBERS.
      ISLIN=59
      ISLOUT=59
      ISLTTY=59
C
C  INITIALIZE THE NUMBER OF BITS PER WORD AND THE OFFSET
      NMBITSW=64
      NRIGHTOF=7
C ------------------- UPDATE #6 -------------------
C
C *** SET PROGRAM VERSION.
      GFTVER=V4B
C ----------------- MWM  11/22/77 -----------------
      RETURN
      END
```

3.8 SUBROUTINE INRA

The function of subroutine INRA is to read a physical record from a direct access file defined on a particular FORTRAN logical unit number and to place the physical record into the appropriate file buffer. This subprogram performs the inverse function of subroutine OUTRA.

Calling sequence:

```
CALL INRA (ISL,NR,IBUFF,NWORDS)
```

Arguments:

ISL	is the FORTRAN logical unit number associated with the direct access file to be read.
NR	is the physical record number of the file to be read.
IBUFF	is the address of the first location in the file's physical record buffer where the physical record will be transferred.
NWORDS	is the number of words to be transferred to the physical record buffer.

Language: CFT-FORTRAN

Subroutine listing:

```
C
C -------------------- NEW ROUTINE (LIBRARY) --------------------
      SUBROUTINE INRA (ISL,NR,IBUFF,NWORDS)
C *** WRITTEN NOVEMBER 23, 1977, BY MICHAEL W. MCCABE
C
C     THIS SUBROUTINE READS 'NWORDS' WORDS INTO ARRAY 'IBUFF' FROM
C     RECORD NUMBER 'NR' OF THE FILE CURRENTLY OPENED ON LOGICAL UNIT
C     NUMBER 'ISL'.
C
      DIMENSION IBUFF(1)
C
      COMMON  /CMDSLT/        ISLIN , ISLOUT , ISLTTY
C
C \/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
C \/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
C \/\                                                               \/\
C \/\ THE ENCLOSED CODE IS DEPENDENT UPON THE COMPUTER SYSTEM BEING \/\
C \/\ USED.  THIS IS THE VERSION TO BE USED ON A CRAY-1 COMPUTER.   \/\
C \/\                                                               \/\
      COMMON /FILBUFT0/       ISLISR(5,2)
C \/\                                                               \/\
C \/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
C \/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
C
```

```
C
C *** READ THE DATA.
C
C \/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
C \/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
C \/\                                                                          \/\
C \/\ THE ENCLOSED CODE IS DEPENDENT UPON THE COMPUTER SYSTEM BEING             \/\
C \/\ USED.  THIS IS THE VERSION TO BE USED ON A CRAY 1 COMPUTER.               \/\
C \/\                                                                          \/\
C
C  CALCULATE THE ZERO-ORIGIN DISK WORD ADDRESS
      IPOSN=(NR-1)*ISLISR(ISL-15,2)
C
C  MUST BE GREATER THAN ZERO
      IF (NR .LE. Ø) GO TO 1Ø5
      ASSIGN 1ØØ TO IERROR
C
C  DO THE READ
      CALL RDABS (ISL,IBUFF,NWORDS,IPOSN,IERROR)
C
      RETURN
C
C  ERROR BRANCH
1ØØ   CONTINUE
      WRITE(ISLTTY,1ØØØ) ISLISR(ISL-15,1),ISL
1ØØØ  FORMAT('?INRA/GIFTS:  ERROR READING DA FILE',A4,' ON CHANNEL',
     1  I5)
      STOP
1Ø5   CONTINUE
C
C  WARNING BRANCH
      WRITE(ISLTTY,1ØØ5) ISLISR(ISL-15,1),ISL,NR
1ØØ5  FORMAT('%FILE -',A4,' , CHANNEL ',I3,' HAS BAD RECORD #',
     1  I5,/,'INPUT OF THIS RECORD HAS BEEN IGNORED')
      DO 11Ø I=1,NWORDS
11Ø   IBUFF(I)=Ø
      RETURN
C \/\                                                                          \/\
C \/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
C \/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
      END
```

3.9 SUBROUTINE OPENIF

Opening a sequential file for input on a particular FORTRAN logical unit number is the function of subroutine OPENIF.

Calling sequence:

CALL OPENIF (ISL,XEXT)

Arguments:

ISL	is the FORTRAN logical unit number associated with the file to be opened.
XEXT	is the extension of the sequential file to be opened.

Language: CFT-FORTRAN

Subroutine listing:

```
C
        SUBROUTINE OPENIF (ISL,EXT)
C
C \/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
C \/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
C \/\                                                               \/\
C \/\ THE ENCLOSED CODE IS DEPENDENT UPON THE COMPUTER SYSTEM BEING \/\
C \/\ USED.  THIS IS THE VERSION TO BE USED ON A CRAY-1 COMPUTER.   \/\
C \/\                                                               \/\
C *** UPDATED: 22JAN79 WHG ***
C
C     THIS SUBROUTINE OPENS SEQUENTIAL FILE JOB,EXT FOR INPUT.
C
        INTEGER  EXT , XJOB , XJOB2
C
        COMMON /FILBUFIO/     ISLISR(5,2)
        COMMON /JOB/XJOB,XJOB2
C
        XJOB2=EXT
C
        CALL PERMN(XJOB,FLNM)
        LEN=0
        CALL OPEN (ISL,FLNM,0,LEN)
        J=ISL-15
        ISLISR(J,1)=EXT
        ISLISR(J,2)=6HOPENIF
C
C \/\                                                               \/\
C \/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
C \/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
        RETURN
        END
```

3.1Ø SUBROUTINE OPENOF

The function of subroutine OPENOF is to open a sequential file for output on a particular FORTRAN logical unit number. The NMFECC CRAY-1 operating system does not allow for two files with the same name to be opened simultaneously. Therefore, subroutine OPENOF changes the extension of the GIFTS estimation file GIFT.EST to GIFT.ES before opening this file for sequential output. When the file is closed, the name will be changed back to GIFT.EST by subroutine CLOSEF.

Furthermore, a special file attribute must be assigned to a file before the text editors available on the NMFECC CRAY-1 are capable of accessing and editing a file. The file extensions of GEO, DSP, DMP, SAP, LPT, and LST are checked for and are created with the text file attribute instead of a sequential file attribute. These extensions are special to the ORNL version of GIFTS. The first two are used by program GFTMOV.[10] The next two are used by GFTDMP[11] and GFTSAP[12], respectively. The last two are reserved for future modification and expansion of GIFTS capability.

Calling sequence:

```
CALL OPENOF (ISL,XEXT)
```

Arguments:

ISL — is the FORTRAN logical unit number associated with the file to be opened.

XEXT — is the extension of the sequential file to be opened.

Language: CFT-FORTRAN

Subroutine listing:

```
C
      SUBROUTINE OPENOF (ISL,EXT)
C
C \/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
C \/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
C \/\                                                                         \/\
C \/\ THE ENCLOSED CODE IS DEPENDENT UPON THE COMPUTER SYSTEM BEING            \/\
C \/\ USED.  THIS IS THE VERSION TO BE USED ON A CRAY-1 COMPUTER.              \/\
C \/\                                                                         \/\
C *** UPDATED: 22JAN79 WHG ***
C
C     THIS SUBROUTINE OPENS SEQUENTIAL FILE JOB,EXT FOR OUTPUT.
C
```

```
      INTEGER  EXT , XJOB , XJOB2 , TEXTFL(6)
C
      COMMON /FILBUFIO/      ISLISR(5,2)
      COMMON /JOB/XJOB,XJOB2
C
      DATA    TEXTFL/ 4H GEO, 4H DSP, 4H DMP, 4H SAP, 4H LPT, 4HLST /
      DATA    NTEXTFL/6/
C
C  IS THIS THE EST FILE
      IF (EXT .EQ. 4H EST) EXT=4H ES
C
      XJOB2=EXT
C
      CALL PERMN(XJOB,FLNM)
C
C  DETERMINE IF SEQUENTIAL OR TEXT FILE TYPE
      ITYPE=1
      DO 1ØØ I=1,NTEXTFL
      IF (EXT .EQ. TEXTFL(I)) ITYPE=2
1ØØ   CONTINUE
      J=ISL-15
      ISLISR(J,1)=EXT
      ISLISR(J,2)=6HOPENOF
      LEN=2ØØØØØB
      CALL CREATE (ISL,FLNM,ITYPE,LEN)
C
C \/\                                                                  \/\
C \/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
C \/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
      RETURN
      END
```

3.11 SUBROUTINE OUTRA

The function of subroutine OUTRA is to write a physical record to a direct access file defined on a particular FORTRAN logical unit number from the appropriate file buffer. This subprogram performs the inverse function of subroutine INRA.

Calling sequence:

```
CALL OUTRA (ISL,NR,IBUFF,NWORDS)
```

Arguments:

ISL	is the FORTRAN logical unit number associated with the direct access file to be written.
NR	is the physical record number of the file to be written.
IBUFF	is the address of the first location in the file's

physical record buffer where the physical record will be transferred.

NWORDS is the number of words to be transferred to the physical record buffer.

Language: CFT-FORTRAN

Subroutine listing:

```
C
C -------------------- NEW ROUTINE (LIBRARY) --------------------
      SUBROUTINE OUTRA (ISL,NR,IBUFF,NWORDS)
C *** WRITTEN NOVEMBER 23, 1977, BY MICHAEL W. MCCABE
C
C     THIS SUBROUTINE WRITES 'NWORDS' WORDS FROM ARRAY 'IBUFF' OUT TO
C     RECORD NUMBER 'NR' OF THE FILE CURRENTLY OPENED ON LOGICAL UNIT
C     NUMBER 'ISL'.
C
      DIMENSION IBUFF(1)
C
      COMMON  /CMDSLT/        ISLIN , ISLOUT , ISLTTY
C
C \/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
C \/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
C \/\                                                                      \/\
C \/\ THE ENCLOSED CODE IS DEPENDENT UPON THE COMPUTER SYSTEM BEING         \/\
C \/\ USED.  THIS IS THE VERSION TO BE USED ON A CRAY-1 COMPUTER.           \/\
C \/\                                                                      \/\
      COMMON /FILBUFIO/       ISLISR(5,2)
C \/\                                                                      \/\
C \/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
C \/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
C
C
C *** WRITE THE DATA.
C
C \/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
C \/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
C \/\                                                                      \/\
C \/\ THE ENCLOSED CODE IS DEPENDENT UPON THE COMPUTER SYSTEM BEING         \/\
C \/\ USED.  THIS IS THE VERSION TO BE USED ON A CRAY-1 COMPUTER.           \/\
C \/\                                                                      \/\
C
C  CALCULATE THE ZERO-ORIGIN DISK WORD ADDRESS
      IPOSN=(NR-1)*ISLISR(ISL-15,2)
      IF (NR .LE. Ø) GO TO 1Ø5
      ASSIGN 1ØØ TO IERROR
C
C  DO THE WRITE
      CALL WRABS (ISL,IBUFF,NWORDS,IPOSN,IERROR)
C
      RETURN
```

```
C
C ERROR BRANCH
1ØØ     CONTINUE
        WRITE(ISLTTY,1ØØØ) ISLISR(ISL-15,1),ISL
1ØØØ    FORMAT('?OUTRA/GIFTS:  ERROR READING DA FILE',A4,' ON CHANNEL',
     1  I5)
        STOP
C
C  WARNING BRANCH
1Ø5     CONTINUE
        WRITE(ISLTTY,1ØØ5) ISLISR(ISL-15,1),ISL,NR
1ØØ5    FORMAT('%FILE -',A4,' , CHANNEL ',I3,' HAS BAD RECORD #',
     1  I5,/,'OUTPUT OF THIS RECORD HAS BEEN IGNORED')
        RETURN
C \/\                                                                  \/\
C \/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
C \/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
        END
```

3.12 SUBROUTINE RENAME

The function of subroutine RENAME is to rename a direct access disk file and to update the XJOB.FIL file accordingly.

Calling sequence:

```
CALL RENAME (EXTOLD,EXTNEW)
```

Arguments:

EXTOLD is the old extension of the direct access file as it currently exists in the user's disk area.

EXTNEW is the new extension of the direct access file to be created in the user's disk area.

Language: CFT-FORTRAN

Subroutine listing:

```
C
        SUBROUTINE RENAME (EXTOLD,EXTNEW)
C *** LAST UPDATED 5/13/78 BY MWM
C *** WRITTEN JANUARY 6, 1976, BY MICHAEL W. MCCABE
C
C     THIS SUBROUTINE RENAMES DISK FILE JOB,EXTOLD TO
C     JOB,EXTNEW, AND MODIFIES THE EXTENSION STORED IN
C     FILE JOB,FIL TO REFLECT THE CHANGE.
C
        LOGICAL FILDEF,PRESNT
C
        DIMENSION FN(2)
        COMMON /DISKIO/INUSE(5),NSF
```

```
        COMMON /JOB/XJOB,XJOB2
        COMMON /FILBUF/FB,LFSZ(5)
C
C
C *** CHECK FOR PRESENCE OF FILE JOB,EXTOLD ON DISK.
C     ERROR IF NOT FOUND.
        IF (.NOT. PRESNT (EXTOLD)) STOP 701
C
C *** CHECK FOR PRESENCE OF FILE JOB,EXTNEW ON DISK.
C     ERROR IF FOUND.
        IF (PRESNT (EXTNEW)) STOP 702
C
C *** CHECK FOR JOB,EXTNEW SPECIFICATIONS IN JOB,FIL.
C     ERROR IF FOUND.
        CALL SLTASN (ISL)
        IF (FILDEF (ISL,EXTNEW,N,LFSZ)) STOP 704
C -------------------- UPDATE #7 --------------------
        CALL CLOSEF (ISL)
C ------------------- MWM 5/13/78 -------------------
C
C *** CHECK FOR JOB,EXTOLD SPECIFICATIONS IN JOB,FIL.
C     ERROR IF NOT FOUND.
        IF (.NOT. FILDEF (ISL,EXTOLD,N,LFSZ)) STOP 703
C
C *** OUTPUT NEW EXTENSION TO JOB,FIL.
        FB=EXTNEW
        CALL OUTRA (ISL,N,FB,NSF)
        CALL CLOSEF (ISL)
C
C *** RENAME DISK FILE.
        XJOB2=EXTNEW
        CALL PERMN (XJOB,NEWFN)
        FN(1)=XJOB
        FN(2)=EXTOLD
        CALL PERMN (FN,OLDFN)
C
C \/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
C \/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
C \/\                                                               \/\
C \/\ THE ENCLOSED CODE IS DEPENDENT UPON THE COMPUTER SYSTEM BEING \/\
C \/\ USED.  THIS IS THE VERSION TO BE USED ON A CRAY-1 COMPUTER.   \/\
C \/\                                                               \/\
        CALL CHGNAME (OLDFN,NEWFN)
C \/\                                                               \/\
C \/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
C \/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
C
C *** RELEASE SLOT 'ISL'.
        CALL FRESLT (ISL)
        RETURN
        END
```

3.13 SUBROUTINE SLTASN

The function of subroutine SLTASN is to assign a FORTRAN logical unit number for further use and to flag its use in array INUSE.

Calling sequence:

CALL SLTASN (ISL)

Arguments:

ISL is the FORTRAN logical unit to be assigned.

Language: CFT-FORTRAN

Subroutine listing:

```
C
      SUBROUTINE SLTASN (ISL)
C  ***   LAST UPDATED 27SEP76   WHG   ***
C *** UPDATED 1/26/76 BY MWM
C *** WRITTEN NOVEMBER 14, 1975, BY MICHAEL W. MCCABE
C
C    THIS SUBROUTINE CHECKS THE DISK LOGICAL UNIT NUMBERS
C    FOR A FREE NUMBER AND RETURNS THAT AS 'ISL'.
C
      DIMENSION LSTLUN(5)
      COMMON /DISKIO/INUSE(5)
      COMMON/CMDSLT/ISLIN,ISLOUT,ISLTTY
C
C
C \/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
C \/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
C \/\                                                               \/\
C \/\ THE ENCLOSED CODE IS DEPENDENT UPON THE COMPUTER SYSTEM BEING \/\
C \/\ USED.  THIS IS THE VERSION TO BE USED ON A CRAY-1 COMPUTER.   \/\
C \/\                                                               \/\
      DATA LSTLUN/16,17,18,19,20/
C \/\                                                               \/\
C \/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
C \/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
C
C *** CHECK FOR FREE SLOT.
      DO 5 I=1,5
      IF (INUSE(I) .EQ. 0) GO TO 10
5     CONTINUE
C
C *** NONE FREE -- STOP.
      WRITE(ISLTTY,1000)
1000  FORMAT(' ?SLTASN/GIFTS - NO FREE SLOTS')
      STOP
C
```

```
C *** FREE SLOT FOUND.  SET INDICATOR TO 'OCCUPY' IT AND RETURN
C     LOGICAL UNIT NUMBER ASSOCIATED WITH IT.
1Ø      ISL=LSTLUN(I)
        INUSE(I)=1
        RETURN
        END
```

4. NMFECC CRAY-1 GIFTS ENCODING/DECODING SOFTWARE

The following sections present the GIFTS system library subroutines that encode and decode words into ASCII character strings. These subroutines have been written in CFT-FORTRAN, and they must be loaded with FORTLIB[3] and BASELIB[4] to satisfy their external references. The following is a brief summary of the differences between the software described in this section and the LIB5 GIFTS library documentation in Ref. 1:

(1) The majority of the subprograms in this section require the subroutine DC8T4 (see Sect. 8.1) to translate true ASCII text strings (A8) into GIFTS ASCII text strings (2A4).

(2) Subroutine PERCNT will write up to five characters for a percentage instead of four. Furthermore, subroutine PERCNT requires subprogram ENIPCT to encode the integer percentage into an ASCII text string.

4.1 SUBROUTINE DATEP

The function of subroutine DATEP is to obtain the current date from the system, reformat the date into a 3A4 format, and return the reformatted date.

Calling sequence:

CALL DATEP (DAT)

Arguments:

DAT	is the current date (returned) formatted into a 3A4 format.

Language: CFT-FORTRAN

Subroutine listing:

```
C
      SUBROUTINE DATEP (ID)
C
C \/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
C \/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
C \/\                                                                    \/\
C \/\ THE ENCLOSED CODE IS DEPENDENT UPON THE COMPUTER SYSTEM BEING \/\
C \/\ USED.  THIS IS THE VERSION TO BE USED ON A CRAY-1 COMPUTER.   \/\
C \/\                                                                    \/\
```

```
C
      DIMENSION ID(1) , A(3)
C
C
C  GET THE SYSTEM DATE
      IF (IZMØ1 (A) .NE. Ø) 999,1Ø
1Ø    CONTINUE
C
C  CONVERT FROM A8 TO 2A4
      CALL DC8T4 (A(2),ID)
      ID(3)=Ø
999   CONTINUE
C \/\                                                                  \/\
C \/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
C \/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
      RETURN
      END
```

4.2 SUBROUTINE ENF

The function of subroutine ENF is to translate a real variable (one floating-point number) into its scientific notation representation in 3A4 format.

Calling sequence:

```
CALL ENF (V,T)
```

Arguments:

V	is the floating-point number to be translated.
T	is the encoded text string with a 3A4 format. The encoding format will produce a scientific notation representation of variable V; i.e., a 1PE1Ø.3 format will be used.

Language: CFT-FORTRAN

Subroutine listing:

```
C
      SUBROUTINE ENF (V,T)
C
C \/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
C \/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
C \/\                                                                  \/\
C \/\ THE ENCLOSED CODE IS DEPENDENT UPON THE COMPUTER SYSTEM BEING     \/\
C \/\ USED.  THIS IS THE VERSION TO BE USED ON A CRAY-1 COMPUTER.       \/\
C \/\                                                                  \/\
C
      DIMENSION T(1) , DUM(2)
```

```
C
C  CONVERT FROM FLOATING POINT TO ASCII E FORMAT REPRESENTATION
        CALL ZCETOA (DUM,Ø,V,1Ø,3)
C
C  CONVERT FROM A8,A2 TO 2A4,A2
        CALL ZMOVECHR (T,Ø,DUM,Ø,4)
        CALL ZMOVECHR (T,8,DUM,4,4)
        CALL ZMOVECHR (T,16,DUM,8,2)
C \/\                                                                    \/\
C \/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
C \/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
        RETURN
        END
```

4.3 SUBROUTINE ENI

The function of subroutine ENI is to encode an integer into a text string (left justified, zero filled) and return the number of characters required by the text string to represent the integer. If more than four characters are required and the integer is positive, then the text string returned will be '****'. If more than three characters are required and the integer is negative, then the text string returned will be '-***'.

Calling sequence:

```
CALL ENI (IN,TEXT,N)
```

Arguments:

IN	is the integer to be encoded.
TEXT	is the ASCII text string (returned) with an A4 format.
N	is the number of characters in the text string.

Language: CFT-FORTRAN

Subroutine listing:

```
C
        SUBROUTINE ENI (I,T,N)
C
        INTEGER T
C
C  BRANCH OF SIZE OF I
        IF (I .LE. 9999 .AND. I .GE. -999) GO TO 1Ø
        N=-4
        T=4H****
        IF (I .LT. Ø) T=4H-***
        RETURN
C
```

```
10      CONTINUE
        IF (I .LE. 999 .AND. I .GE. -99) GO TO 20
        N=4
        GO TO 50
C
20      CONTINUE
        IF (I .LE. 99 .AND. I .GE. -9) GO TO 30
        N=3
        GO TO 50
C
30      CONTINUE
        IF (I .LE. 9 .AND. I .GE. 0) GO TO 40
        N=2
        GO TO 50
C
40      CONTINUE
        N=1
50      CONTINUE
        T=0
C
C \/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
C \/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
C \/\                                                              \/\
C \/\ THE ENCLOSED CODE IS DEPENDENT UPON THE COMPUTER SYSTEM BEING \/\
C \/\ USED.  THIS IS THE VERSION TO BE USED ON A CRAY-1 COMPUTER.   \/\
C \/\                                                              \/\
        CALL ZCITOA (T,0,I,N,0)
C \/\                                                              \/\
C \/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
C \/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
C
1000    FORMAT(I4)
1005    FORMAT(I3)
1010    FORMAT(I2)
1015    FORMAT(I1)
C
        END
```

4.4 SUBROUTINE ENIPCT

The function of subroutine ENIPCT is to encode an integer into a text string, followed by a percent (%) sign, and return the number of characters required to represent the integer and the percent sign. If more than five characters are required and IS is positive, then the text string returned will be '****%'. If more than four characters are required and IS is negative, then the text string returned will be '-***%'. The format of the text string is 2A4. This subroutine is a

special-purpose subprogram called only by the NMFECC CRAY-1 version of PERCNT.

Calling sequence:

```
     CALL ENIPCT (IN,IS,TEXT,N)
```

Arguments:

IN	is the integer to be encoded.
IS	is the sign of the integer IN so that '-0%' can be distinguished from '0%'.
TEXT	is the ASCII text string (returned) with a 2A4 format.
N	is the number of characters returned in the text string.

Language: CFT-FORTRAN

Subroutine listing:

```
C
      SUBROUTINE ENIPCT (I,IS,T,N)
C
C \/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
C \/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
C \/\                                                                \/\
C \/\ THE ENCLOSED CODE IS DEPENDENT UPON THE COMPUTER SYSTEM BEING \/\
C \/\ USED.  THIS IS THE VERSION TO BE USED ON A CRAY-1 COMPUTER.   \/\
C \/\                                                                \/\
C
      DIMENSION T(1)
C
C  BRANCH ON IS
      IF (IS .LT. 0) GO TO 50
C
C  BRANCH ON SIZE OF I
      IF (I .LE. 9999) GO TO 10
      N=5
      J=5H****%
      GO TO 200
C
10    CONTINUE
      IF (I .LE. 999) GO TO 20
      N=5
      GO TO 100
C
20    CONTINUE
      IF (I .LE. 99) GO TO 30
      N=4
      GO TO 100
C
30    CONTINUE
      IF (I .LE. 9) GO TO 40
      N=3
```

```
        GO TO 1ØØ
C
4Ø      CONTINUE
        N=2
        GO TO 1ØØ
5Ø      CONTINUE
C
C  BRANCH OF SIZE OF I
        IF (I .LE. 999) GO TO 6Ø
        N=5
        J=5H-***%
        GO TO 2ØØ
C
6Ø      CONTINUE
        IF (I .LE. 99) GO TO 7Ø
        N=5
        GO TO 1ØØ
C
7Ø      CONTINUE
        IF (I .LE. 9) GO TO 8Ø
        N=4
        GO TO 1ØØ
C
8Ø      CONTINUE
        N=3
1ØØ     CONTINUE
        J=Ø
        IF (IS .LT. Ø) GO TO 11Ø
C
C  CONVERT FROM INTEGER TO ASCII
        CALL ZCITOA (J,Ø,I,N-1,Ø)
        GO TO 12Ø
11Ø     CONTINUE
C
C  SPECIAL CASE FOR NEGATIVE IS
        J=1H-
C
C  CONVERT FROM INTEGER TO ASCII
        CALL ZCITOA (J,1,I,N-2,Ø)
12Ø     CONTINUE
C
C  GET THE % SIGN
        CALL ZMOVECHR (J,N-1,1H%,Ø,1)
2ØØ     CONTINUE
C
C  CONVERT FROM A8 TO 2A4
        CALL DC8T4 (J,T)
        RETURN
C \/\                                                                    \/\
C \/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
C \/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
        END
```

4.5 SUBROUTINE INCCHR

The function of subroutine INCCHR is to alphabetically increment a left-justified ASCII character. For example, the letter A will become the letter B, ... This subroutine is used in the contour plotting sections of the GIFTS system.

Calling sequence:

CALL INCCHR (CHR)

Arguments:

CHR	is the character to be incremented. The incremented character is returned in CHR.

Language: CFT-FORTRAN

Subroutine listing:

```
C
      SUBROUTINE INCCHR (I)
C
C \/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
C \/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
C \/\                                                                  \/\
C \/\ THE ENCLOSED CODE IS DEPENDENT UPON THE COMPUTER SYSTEM BEING \/\
C \/\ USED.  THIS IS THE VERSION TO BE USED ON A CRAY-1 COMPUTER.   \/\
C \/\                                                                  \/\
C
      DATA    IA/     0004000000000000000000B/
C \/\                                                                  \/\
C \/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
C \/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
C
      I=I+IA
      RETURN
      END
```

4.6 SUBROUTINE INCNM

Adding an increment to an alphanumeric name is the function of subroutine INCNM. It is used whenever a series of lines or grids is to be generated. The second word of the alphanumeric name is assumed to be a four-character ASCII representation of an integer. This ASCII text string is decoded from the alphanumeric name into an integer, the

increment is added to it, and it is then encoded back into a four-character text string to be returned in the updated alphanumeric name.

For example, if a name is SIDEØØ5Ø and an increment is 2ØØ, subprogram INCNM will return the name SIDEØ25Ø. Furthermore, in order to prevent overflow of the integer out of the name, the sum of the numeric portion of the name and the increment must be computed as modulo 1ØØØØ. For example, SIDE9999 plus 1 yields SIDEØØØØ.

Calling sequence:

```
CALL INCNM (XNAME,INCR)
```

Arguments:

XNAME	is an alphanumeric name to be incremented (stored in 2A4 format). The second word of XNAME is assumed to be an encoded integer.
INCR	is the increment to be added to XNAME(2).

Language: CFT-FORTRAN

Subroutine listing:

```
      SUBROUTINE INCNM (XNAME,INC)
C
      DIMENSION XNAME(2)
C
C
C \/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
C \/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
C \/\                                                                        \/\
C \/\ THE ENCLOSED CODE IS DEPENDENT UPON THE COMPUTER SYSTEM BEING           \/\
C \/\ USED.  THIS IS THE VERSION TO BE USED ON A CRAY-1 COMPUTER.            \/\
C \/\                                                                        \/\
C
C  FIRST CONVERT ASCII CHARACTERS INTO FIXED-POINT NUMBER
      CALL ZCIATOB (XNAME(2),Ø,IOUT,I,TC)
C \/\                                                                        \/\
C \/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
C \/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
C
C  ADD INCREMENT AND COMPUTE ITS MODULO 1ØØØØ
      I=I+INC
      IF (I .GE. 1ØØØØ) I=I-1ØØØØ
      IF (I .LT. Ø) I=I+1ØØØØ
C
C \/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
C \/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
C \/\                                                                        \/\
```

```
C \/\ THE ENCLOSED CODE IS DEPENDENT UPON THE COMPUTER SYSTEM BEING \/\
C \/\ USED.  THIS IS THE VERSION TO BE USED ON A CRAY-1 COMPUTER.   \/\
C \/\                                                              \/\
C
C  NOW CONVERT BACK TO ASCII
      CALL ZCITOA (XNAME(2),Ø,I,4,4)
C \/\                                                              \/\
C \/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
C \/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
      RETURN
      END
```

4.7 SUBROUTINE NUMCHR

The function of subroutine NUMCHR is to count the number of nonblank characters in an alphanumeric name. The name is stored in a 2A4 format and therefore can have a maximum of eight characters.

Calling sequence:

```
CALL NUMCHR (XNAME,NUMBER)
```

Arguments:

XNAME — is the alphanumeric name that is to have its nonblank characters counted.

NUMBER — is the number of nonblank characters in XNAME.

Language: CFT-FORTRAN

Subroutine listing:

```
C
      SUBROUTINE NUMCHR (NAME,I)
C
C \/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
C \/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
C \/\                                                              \/\
C \/\ THE ENCLOSED CODE IS DEPENDENT UPON THE COMPUTER SYSTEM BEING \/\
C \/\ USED.  THIS IS THE VERSION TO BE USED ON A CRAY-1 COMPUTER.   \/\
C \/\                                                              \/\
C
      DIMENSION NAME(1) , ICHR(4)
C
      DATA    IS/ 1R /
C
      I=Ø
C
C  LOOP OVER BOTH WORDS
      DO 1ØØ J=1,2
C
C  CONVERT A4 ASCII INTO ADE CHARACTERS
```

```
        CALL ZRJCHARZ (ICHR,NAME(J),4)
        DO 1ØØ K=1,4
C
C  CHECK FOR A SPACE
        IF (ICHR(K) .EQ. IS) GO TO 1Ø5
        I=I+1
1ØØ     CONTINUE
1Ø5     CONTINUE
C \/\                                                                    \/\
C \/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
C \/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
        RETURN
        END
```

4.8 SUBROUTINE PAKAL

The function of subroutine PAKAL is to pack N characters that are stored right justified, one character per word, into a left-justified ASCII text string. There can be a maximum of four characters to pack for any single call.

Calling sequence:

```
CALL PAKAL (NUMBER,RIGHTJ,TEXT)
```

Arguments:

NUMBER	is the number of characters to pack.
RIGHTJ	is the array containing the right-justified characters to be packed. The maximum number of possible characters to pack is four. These characters are stored one per word.
TEXT	is the left-justified ASCII text string that will be returned.

Language: CFT-FORTRAN

Subroutine listing:

```
C
        SUBROUTINE PAKAL (N,L,TEXT)
C
C \/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
C \/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
C \/\                                                                    \/\
C \/\ THE ENCLOSED CODE IS DEPENDENT UPON THE COMPUTER SYSTEM BEING       \/\
C \/\ USED.  THIS IS THE VERSION TO BE USED ON A CRAY-1 COMPUTER.         \/\
C \/\                                                                    \/\
C
        DIMENSION L(1)
C
```

```
      DATA    BLANK/  8H         /
C
      TEXT=BLANK
C
C  NOW PACK THE RIGHT-JUSTIFIED CHARACTERS INTO LEFT-JUSTIFIED
C  TEXT
      CALL ZPAKCHRZ (TEXT,L,N)
      TEXT=TEXT.OR.BLANK
C \/\                                                                \/\
C \/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
C \/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
      RETURN
      END
```

4.9 SUBROUTINE PERCNT

The function of subroutine PERCNT is to plot an integer number followed by a percent sign at a specified location on the screen of a Tektronix storage tube. This subprogram uses subroutine ENIPCT to encode the integer into a text string and then subroutine TEXT to plot the resulting percentage. The sign of the integer is specified in order that '-Ø%' and 'Ø%' can be distinguished.

Calling sequence:

```
CALL PERCNT (IN,IS,IXPOS,IYPOS,IOFFH,IOFFV)
```

Arguments:

IN — is the integer to be encoded and plotted as a percentage.

IS — is the sign of integer IN. This argument allows subroutine PERCNT to distinguish between '-Ø%' and 'Ø%'.

IXPOS — is the integer x-coordinate where the percentage is to be plotted.

IYPOS — is the integer y-coordinate where the percentage is to be plotted.

IOFFH — is a horizontal offset to be applied to the x-coordinate before plotting the percentage. If IOFFH is equal to -1, then the percentage is to be offset to the left. If IOFFH is equal to Ø, then the percentage is to be horizontally centered. If IOFFH is equal to 1, then the percentage is to be offset to the right.

IOFFV — is a vertical offset to be applied to the y-coordinate

before plotting the percentage. If IOFFV is equal to -1, then the percentage is to be offset to the bottom. If IOFFV is equal to Ø, then the percentage is to be vertically centered. If IOFFV is equal to 1, then the percentage is to be offset to the top.

Language: CFT-FORTRAN

Subroutine listing:

```
C
      SUBROUTINE PERCNT (I,IS,IX,IY,IDX,IDY)
C
C  SUBROUTINE PERCNT
C
C  PURPOSE:
C
C THIS SUBROUTINE ENCODES AN INTEGER I, FOLLOWED BY A PERCENT
C  SIGN, INTO THE FLOATING-POINT VARIABLE (T) (LEFT-JUSTIFIED,
C  ZERO FILLED) AND STORES THE REQUIRED NUMBER OF
C  CHARACTERS IN (N).
C  IF MORE THAN 5 CHARACTERS ARE REQUIRED, (T) WILL BE ****%
C  OR -***% , AND N WILL BE 5.
C  THE SIGN OF THE DISPLAYED PERCENTAGE IS PASSED AS (IS) (+1,-1);
C  (I) IS ALWAYS POSITIVE.
C THIS IS TO DISTINGUISH BETWEEN +Ø AND -Ø.
C
C  WHEN ENCODING IS COMPLETE,  THE CHARACTER STRING IN (T) WILL
C  BE PLOTTED AT THE SCREEN COORDINATES (IX,IY),  WITH THE OFFSET
C  SPECIFIED BY:
C
C       IDX = -1    OFFSET LEFT
C       IDX =  Ø    CENTERED HORIZONTALLY
C       IDX =  1    OFFSET RIGHT
C       IDY = -1    OFFSET DOWN
C       IDY =  Ø    CENTERED VERTICALLY
C       IDY =  1    OFFSET UP
C
C  AUTHOR:      W. H. GRAY
C               FUSION ENERGY DIVISION
C               OAK RIDGE NATIONAL LABORATORY
C               OAK RIDGE, TENNESSEE 3783Ø
C
C  SUBROUTINES AND FUNCTIONAL SUBPROGRAMS REQUIRED:
C
C       ENIPCT  -       GIFTS LIBRARY
C       TEXT    -       GIFTS LIBRARY
C       SETPT   -       GIFTS LIBRARY
C
      DIMENSION T(2)
C
```

```
C *** ENCODE 'I' AND GET NUMBER OF CHARACTERS (N)
        CALL ENIPCT(I,IS,T,N)
C
C *** DETERMINE SCREEN COORDINATES (IXC,IYC) OF LOWER-LEFT
C     CORNER OF TEXT STRING
        NRU=7*N
        IF (IDX) 5,1Ø,15
5       IXC=IX-(NRU+NRU+1)
        GO TO 2Ø
1Ø      IXC=IX-NRU+2
        GO TO 2Ø
15      IXC=IX+5
C
2Ø      IF (IDY) 25,3Ø,35
25      IYC=IY-19
        GO TO 4Ø
3Ø      IYC=IY-7
        GO TO 4Ø
35      IYC=IY+5
C
C *** PLOT THE ENCODED INTEGER
4Ø      CALL SETPT (IXC,IYC)
        CALL TEXT (T,N)
        RETURN
        END
```

4.10 SUBROUTINE PERMN

The function of subroutine PERMN is to create a file name from the job name and a file extension. The job name can have a maximum of four nonblank characters that are stored left justified. The file extension has three characters and occupies the second-through-fourth character positions in the second word of the job name array. The job name array is the first argument of this subprogram. The resulting file name is a true ASCII text string with an A8 format. A period is inserted between the job name and the file extensions.

Calling sequence:

```
CALL PERMN (XJOB,FLNM)
```

Arguments:

XJOB	is the job name (XJOB(1)) and the file extension (XJOB(2)) stored in the special format described above.
FLNM	is the file name created by this subprogram.

Language: CFT-FORTRAN

Subroutine listing:

```
C
      SUBROUTINE PERMN (IJOBN,INAME)
C
C \/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
C \/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
C \/\                                                                          \/\
C \/\ THE ENCLOSED CODE IS DEPENDENT UPON THE COMPUTER SYSTEM BEING             \/\
C \/\ USED.  THIS IS THE VERSION TO BE USED ON A CRAY-1 COMPUTER.               \/\
C \/\                                                                          \/\
C
      DIMENSION IJOBN(1) , JJOB(4)
C
      DATA    PERIOD/ 1H./
C
C
C  DECODE THE JOB NAME INTO ADE CHARACTERS
      CALL ZRJCHARZ (JJOB,IJOBN,4)
      I=Ø
C
C  SEARCH FOR A PERIOD
      DO 1ØØ J=1,4
      IF (JJOB(J) .EQ. 32 .OR. JJOB(J) .EQ. Ø) GO TO 1Ø5
      I=I+1
1ØØ   CONTINUE
1Ø5   CONTINUE
      INAME=Ø
C
C  PLACE THE JOB NAME INTO THE FILE NAME
      CALL ZMOVECHR (INAME,Ø,IJOBN(1),Ø,I)
C
C  PLACE A PERIOD INTO THE FILE NAME
      CALL ZMOVECHR (INAME,I,PERIOD,Ø,1)
C
C  PLACE AN EXTENSION INTO THE FILE NAME
      CALL ZMOVECHR (INAME,I+1,IJOBN(2),1,3)
C \/\                                                                          \/\
C \/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
C \/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
      RETURN
      END
```

4.11 SUBROUTINE TIMEP

The function of subroutine TIMEP is to obtain the current time from the system, reformat the time into a 3A4 format, and return the reformatted time.

Calling sequence:

    CALL TIMEP (TIM)

Arguments:

TIM	is the current time (returned) formatted into a 3A4 format.

Language: CFT-FORTRAN

Subroutine listing:

```
C
        SUBROUTINE TIMEP (IT)
C
C \/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
C \/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
C \/\                                                               \/\
C \/\ THE ENCLOSED CODE IS DEPENDENT UPON THE COMPUTER SYSTEM BEING \/\
C \/\ USED.  THIS IS THE VERSION TO BE USED ON A CRAY-1 COMPUTER.   \/\
C \/\                                                               \/\
C
        DIMENSION IT(1) , A(3)
        A(1)=0.
C
C  GET THE SYSTEM TIME
        IF (IZM01 (A) .NE. 0) 999,10
10      CONTINUE
C
C  CONVERT FROM A8 TO 2A4
        CALL DC8T4 (A,IT)
C
C  ADD A C TO INDICATE THAT WE ARE RUNNING ON THE CRAY
        A(2)=1R
        CALL ZPAKCHRZ (IT(3),A(2),2)
999     CONTINUE
C \/\                                                               \/\
C \/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
C \/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
        RETURN
        END
```

5. NMFECC CRAY-1 GIFTS BIT PACKING/UNPACKING SOFTWARE

In order that the XJOB.PTS file not be unnecessarily large, the nodal freedom pattern and prescribed displacement binary switches are packed into a single word prior to inclusion in the XJOB.PTS logical record. (The XJOB.PTS file contains the nodal point information for all the points in a particular model.) This bit packing effects a twelve-to-one word space reduction. Also provided are subroutines to unpack the values back into twelve individual words so that the rest of the FORTRAN code can handle these binary data. The packing subroutine is named PKF, and the unpacking subroutines are named UPF and UPF1. These three subprograms are described in the next sections.

5.1 SUBROUTINE PKF

The function of subroutine PKF is to pack the nodal freedom and prescribed displacement pattern from a twelve-word array into a single word.

Calling sequence:

CALL PKF (LFREE,MPACK)

Arguments:

LFREE	is a twelve-word array containing 0's and 1's.
MPACK	is a single-integer word, the twelve left-most bits of which represent the 0's and 1's of the nodal freedom array. This subprogram packs LFREE into MPACK.

Language: CFT-FORTRAN

Subroutine listing:

```
C
        SUBROUTINE PKF (L,MFP)
C
C \/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
C \/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
C \/\                                                               \/\
C \/\ THE ENCLOSED CODE IS DEPENDENT UPON THE COMPUTER SYSTEM BEING \/\
C \/\ USED.  THIS IS THE VERSION TO BE USED ON A CRAY-1 COMPUTER.   \/\
C \/\                                                               \/\
C
        DIMENSION L(1)
C
```

```
      COMMON  /MACHDPN/       NMBITSW , NRIGHTOF
C
      NBO=NMBITSW-1
      MFP=Ø
      DO 1ØØ I=1,12
      IBITPOS=I-1
C
C  TRANSFER A BIT FROM L(I) TO MFP
      CALL ZMOVEBIT (MFP,IBITPOS,L(I),NBO,1)
1ØØ   CONTINUE
C \/\                                                                  \/\
C \/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
C \/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
      RETURN
      END
```

5.2 SUBROUTINE UPF

The function of subroutine UPF is to unpack the nodal freedom and prescribed displacement pattern from a single-integer word into a twelve-word array.

Calling sequence:

```
CALL UPF (MPACK,LFREE)
```

Arguments:

MPACK is a single-integer word, the twelve left-most bits of which represent the Ø's and 1's of the nodal freedom array.

LFREE is a twelve-word array containing Ø's and 1's. This subroutine performs the inverse function of PKF; i.e., subroutine UPF unpacks MPACK into LFREE.

Language: CFT-FORTRAN

Subroutine listing:

```
C
      SUBROUTINE UPF (MFP,L)
C
C \/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
C \/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
C \/\                                                                  \/\
C \/\ THE ENCLOSED CODE IS DEPENDENT UPON THE COMPUTER SYSTEM BEING     \/\
C \/\ USED.  THIS IS THE VERSION TO BE USED ON A CRAY-1 COMPUTER.       \/\
C \/\                                                                  \/\
C
      DIMENSION L(1)
```

```
C
      COMMON  /MACHDPN/       NMBITSW , NRIGHTOF
C
      NBO=NMBITSW-1
      DO 1ØØ I=1,12
      IBITPOS=I-1
      L(I)=Ø
C
C  TRANSFER A BIT FROM MFP TO L(I)
      CALL ZMOVEBIT (L(I),NBO,MFP,IBITPOS,1)
1ØØ   CONTINUE
C \/\                                                                    \/\
C \/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
C \/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
      RETURN
      END
```

5.3 SUBROUTINE UPF1

The function of subroutine UPF1 is to unpack the nodal freedom and prescribed displacement pattern from a single-integer word into a twelve-word array and to return the number of freedoms in LFREE.

Calling sequence:

```
CALL UPF1 (MPACK,LFREE,NFREE)
```

Arguments:

MPACK — is a single-integer word, the twelve left-most bits of which represent the Ø's and 1's of the nodal freedom array.

LFREE — is a twelve-word array containing Ø's and 1's. This subroutine performs the inverse function of PKF; i.e., subroutine UPF1 unpacks MPACK into LFREE.

NFREE — is the total number of freedoms in LFREE. This value is simply the number of 1's in array LFREE.

Language: CFT-FORTRAN

Subroutine listing:

```
      SUBROUTINE UPF1 (MFP,L,N)
C
      DIMENSION L(1)
C
      CALL UPF (MFP,L)
      N=Ø
      DO 1ØØ I=1,12
```

```
      N=N+L(I)
1ØØ   CONTINUE
      RETURN
      END
```

---

6. NMFECC CRAY-1 GIFTS TIMING SOFTWARE

The function of subroutine SECOND is to return the elapsed CPU time since the beginning of program execution. The time is returned in a real variable.

Calling sequence:

CALL SECOND (SECS)

Arguments:

SECS	is the number of seconds elapsed since the beginning of program execution.

Language: CFT-FORTRAN

Subroutine listing:

```
C
      SUBROUTINE SECOND (SECS)
C
C \/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
C \/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
C \/\                                                               \/\
C \/\ THE ENCLOSED CODE IS DEPENDENT UPON THE COMPUTER SYSTEM BEING \/\
C \/\ USED.  THIS IS THE VERSION TO BE USED ON A CRAY-1 COMPUTER.   \/\
C \/\                                                               \/\
C
      DIMENSION I(5)
C
      SECS=0.0
C
C  GET THE TOTAL CPU, IO, AND SYSTEM TIME FOR THIS JOB
      IF (IZM00 (I) .NE. 0) 999,10
10    CONTINUE
C
C  CONVERT TO SECONDS
      SECS=1.0E-6*FLOAT(I(1)+I(2)+I(3))
999   CONTINUE
C \/\                                                               \/\
C \/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
C \/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
      RETURN
      END
```

7. NMFECC CRAY-1 MODIFIED GIFTS SOFTWARE

Two separate modifications are described in this section. The first deals with how the ORNL commands have been implemented into the NMFECC CRAY-1 version of GIFTS. The second discusses a modification to GIFTS subprogram NEWPG to prevent user interaction with GIFTS under certain circumstances.

7.1 GIFTS MODIFIED COMMAND SCANNERS

The NMFECC CRAY-1 version of GIFTS has eight added commands (these added commands are discussed in the next section). A method was chosen to implement these commands so that a minimum of work would be required every time the University of Arizona released an update of its version of the GIFTS software. In each interactive module where inclusion of the NMFECC CRAY-1 commands was deemed necessary, a call to subroutine URSCMD was inserted after the call to subroutine FREAD in the appropriate command handler subroutine of the particular GIFTS module. An example of a modified subroutine — COMBM, the command scanner for module BULKM — follows. (Notice the underlined call to subroutine USRCMD after the call to subroutine FREAD. These calls are approximately the first executable FORTRAN statements in this subprogram.)

Modified subroutine COMBM listing:

```
      SUBROUTINE COMBM
C *** LAST UPDATED 22JAN79 WHG
C *** LAST UPDATED 4/26/78 BY MWM
C *** REWRITTEN JUNE 27, 1975, BY MICHAEL W. MCCABE
C
C     SUBROUTINE TO READ USER COMMANDS FOR PROGRAM 'BULKM'.
C
      LOGICAL REPEAT,IBELL
C
      DIMENSION TF(2,7Ø),T(9Ø),U(5Ø),NA(7Ø),NI(7Ø),NF(7Ø),CMND(2)
      DIMENSION LRPT(4)
      COMMON /CMDSLT/ISLIN,ISLOUT,ISLTTY
      COMMON /JOB/XJOB,XJOB2,IERR,IBELL
      COMMON /COMMND/L(59),MODE,V(2Ø),CMND1,CMND2,NCMND,REPEAT
      COMMON /ELTS/ITL
C
      EQUIVALENCE (L,L1),(L(2),L2),(V,V1),(CMND1,CMND)
```

```
      EQUIVALENCE (TF,T),(TF(1,46),U)
C
      DATA BLANK/1H /, NLCLM/17/, NTHV/15/, NUMCMD/7Ø/
C
C ************************** COMMAND LIST ******************************
C
C                                                  COMMAND       NA  NI  NF
      DATA
     1  T( 1),T( 2),NA( 1),NI( 1),NF( 1)/4HVDIR,4H     ,  Ø,  3,  Ø/
     2 ,T( 3),T( 4),NA( 2),NI( 2),NF( 2)/4HROTV,4H     ,  Ø,  Ø,  1/
     3 ,T( 5),T( 6),NA( 3),NI( 3),NF( 3)/4HBOX ,4H     ,  Ø,  Ø,222/
     4 ,T( 7),T( 8),NA( 4),NI( 4),NF( 4)/4HWIND,4HOW   ,  Ø,  Ø,  Ø/
     5 ,T( 9),T(1Ø),NA( 5),NI( 5),NF( 5)/4HVDIS,4HT    ,  Ø,  Ø,  1/
     6 ,T(11),T(12),NA( 6),NI( 6),NF( 6)/4HGON ,4H     ,1ØØ,  Ø,  Ø/
     7 ,T(13),T(14),NA( 7),NI( 7),NF( 7)/4HGOFF,4H     ,1ØØ,  Ø,  Ø/
     8 ,T(15),T(16),NA( 8),NI( 8),NF( 8)/4HKN  ,4H     ,  Ø,  Ø,  Ø/
     9 ,T(17),T(18),NA( 9),NI( 9),NF( 9)/4HLNAM,4H     ,  Ø,  Ø,  Ø/
     1 ,T(19),T(2Ø),NA(1Ø),NI(1Ø),NF(1Ø)/4HGNAM,4H     ,  Ø,  Ø,  Ø/
     2 ,T(21),T(22),NA(11),NI(11),NF(11)/4HRESE,4HT    ,  Ø,  Ø,  Ø/
     3 ,T(23),T(24),NA(12),NI(12),NF(12)/4HPLOT,4H     ,  Ø,  Ø,  Ø/
     4 ,T(25),T(26),NA(13),NI(13),NF(13)/4HPLOT,4HNL   ,  Ø,  Ø,  Ø/
     5 ,T(27),T(28),NA(14),NI(14),NF(14)/4HOLB ,4H     ,  1,  Ø,  Ø/
     6 ,T(29),T(3Ø),NA(15),NI(15),NF(15)/4HEND ,4H     ,  Ø,  Ø,  Ø/
     7 ,T(31),T(32),NA(16),NI(16),NF(16)/4HBELL,4HON   ,  Ø,  Ø,  Ø/
     8 ,T(33),T(34),NA(17),NI(17),NF(17)/4HBELL,4HOFF  ,  Ø,  Ø,  Ø/
     9 ,T(35),T(36),NA(18),NI(18),NF(18)/4HPAGE,4HON   ,  Ø,  Ø,  Ø/
     1 ,T(37),T(38),NA(19),NI(19),NF(19)/4HPAGE,4HOFF  ,  Ø,  Ø,  Ø/
     2 ,T(39),T(4Ø),NA(2Ø),NI(2Ø),NF(2Ø)/4HQUIT,4H     ,  Ø,  Ø,  Ø/
     3 ,T(41),T(42),NA(21),NI(21),NF(21)/4HJOB ,4H     ,  Ø,  Ø,  Ø/
     4 ,T(43),T(44),NA(22),NI(22),NF(22)/4HELMA,4HT    ,  Ø,1ØØ,  6/
     5 ,T(45),T(46),NA(23),NI(23),NF(23)/4HORMA,4HT    ,  Ø,1ØØ,  9/
     6 ,T(47),T(48),NA(24),NI(24),NF(24)/4HETH ,4H     ,  Ø,1ØØ,  9/
     7 ,T(49),T(5Ø),NA(25),NI(25),NF(25)/4HRECT,4HS    ,  Ø,1ØØ,  2/
     8 ,T(51),T(52),NA(26),NI(26),NF(26)/4HRECT,4HH    ,  Ø,1ØØ, 22/
     9 ,T(53),T(54),NA(27),NI(27),NF(27)/4HCIRC,4HS    ,  Ø,1ØØ,  1/
     1 ,T(55),T(56),NA(28),NI(28),NF(28)/4HCIRC,4HH    ,  Ø,1ØØ,  2/
     2 ,T(57),T(58),NA(29),NI(29),NF(29)/4HGIBE,4HAM   ,  Ø,1ØØ,332/
     3 ,T(59),T(6Ø),NA(3Ø),NI(3Ø),NF(3Ø)/4HIBEA,4HM    ,  Ø,1ØØ, 22/
     4 ,T(61),T(62),NA(31),NI(31),NF(31)/4HANGL,4HE    ,  Ø,1ØØ, 22/
     5 ,T(63),T(64),NA(32),NI(32),NF(32)/4HCHAN,4HNEL  ,  Ø,1ØØ, 22/
     6 ,T(65),T(66),NA(33),NI(33),NF(33)/4HTBEA,4HM    ,  Ø,1ØØ, 22/
     7 ,T(67),T(68),NA(34),NI(34),NF(34)/4HZBEA,4HM    ,  Ø,1ØØ, 22/
     8 ,T(69),T(7Ø),NA(35),NI(35),NF(35)/4HOBAN,4HG    ,  Ø,1ØØ,111/
     9 ,T(71),T(72),NA(36),NI(36),NF(36)/4HMSTE,4HEL   ,  Ø,1ØØ,  6/
     1 ,T(73),T(74),NA(37),NI(37),NF(37)/4HSSTE,4HEL   ,  Ø,1ØØ,  6/
     2 ,T(75),T(76),NA(38),NI(38),NF(38)/4HAALL,4HOY   ,  Ø,1ØØ,  6/
     3 ,T(77),T(78),NA(39),NI(39),NF(39)/4HHELP,4H     ,  Ø,  Ø,  Ø/
     4 ,T(79),T(8Ø),NA(4Ø),NI(4Ø),NF(4Ø)/4HINFM,4HOD   ,  Ø,  Ø,  Ø/
     5 ,T(81),T(82),NA(41),NI(41),NF(41)/4HINFK,4H     ,  Ø,3ØØ,  Ø/
     6 ,T(83),T(84),NA(42),NI(42),NF(42)/4HINFL,4H     ,  Ø,3ØØ,  Ø/
     7 ,T(85),T(86),NA(43),NI(43),NF(43)/4HINFG,4H     ,  Ø,3ØØ,  Ø/
     8 ,T(87),T(88),NA(44),NI(44),NF(44)/4HKPOI,4HNT   ,  Ø,3ØØ, 3Ø/
```

```
     9 ,T(89),T(9Ø),NA(45),NI(45),NF(45)/  Ø.   ,  Ø.   ,  Ø,  Ø,  Ø/
     1 ,U( 1),U( 2),NA(46),NI(46),NF(46)/  Ø.   ,  Ø.   ,  Ø,  Ø,  Ø/
     2 ,U( 3),U( 4),NA(47),NI(47),NF(47)/4HSLIN,4HE    ,1ØØ, 7Ø,  Ø/
     3 ,U( 5),U( 6),NA(48),NI(48),NF(48)/4HCARC,4H     ,1ØØ, 7Ø,  Ø/
     4 ,U( 7),U( 8),NA(49),NI(49),NF(49)/4HPARA,4HM2   ,1ØØ, 7Ø,  Ø/
     5 ,U( 9),U(1Ø),NA(5Ø),NI(5Ø),NF(5Ø)/4HPARA,4HM3   ,1ØØ, 7Ø,  Ø/
     6 ,U(11),U(12),NA(51),NI(51),NF(51)/  Ø.   ,  Ø.   ,1Ø4,  Ø,  1/
     7 ,U(13),U(15),NA(52),NI(52),NF(52)/  Ø.   ,  Ø.   ,1Ø4, 24,  1/
     8 ,U(15),U(16),NA(53),NI(53),NF(53)/4HGRID,4HP    ,1Ø4,  Ø,  Ø/
     9 ,U(17),U(18),NA(54),NI(54),NF(54)/4HGRID,4HPS   ,1Ø4, 24,  Ø/
     1 ,U(19),U(2Ø),NA(55),NI(55),NF(55)/4HGRID,4HC    ,1Ø4,  1,  Ø/
     2 ,U(21),U(22),NA(56),NI(56),NF(56)/4HGRID,4HCS   ,1Ø4, 24,  Ø/
     3 ,U(23),U(24),NA(57),NI(57),NF(57)/4HGRID,4H4    ,1Ø4,  Ø,  Ø/
     4 ,U(25),U(26),NA(58),NI(58),NF(58)/4HGRID,4H4S   ,1Ø4, 24,  Ø/
     5 ,U(27),U(28),NA(59),NI(59),NF(59)/4HGRID,4H3    ,1Ø3,  Ø,  Ø/
     6 ,U(29),U(3Ø),NA(6Ø),NI(6Ø),NF(6Ø)/4HGRID,4H3S   ,1Ø3, 23,  Ø/
     7 ,U(31),U(32),NA(61),NI(61),NF(61)/4HGRID,4HT    ,1Ø2,  Ø,  Ø/
     8 ,U(33),U(34),NA(62),NI(62),NF(62)/4HGRID,4HTS   ,1Ø2, 22,  Ø/
     9 ,U(35),U(36),NA(63),NI(63),NF(63)/4HCOMP,4HGRID,1Ø9,  Ø,  Ø/
     1 ,U(37),U(38),NA(64),NI(64),NF(64)/4HCOMP,4HLINE,1Ø9,  Ø,  Ø/
     2 ,U(39),U(4Ø),NA(65),NI(65),NF(65)/4HLETY,4H     ,  2,  2,  Ø/
     3 ,U(41),U(42),NA(66),NI(66),NF(66)/4HGETY,4H     ,  2,  2,  Ø/
     4 ,U(43),U(44),NA(67),NI(67),NF(67)/4HDELL,4H     ,1ØØ,  Ø,  Ø/
     5 ,U(45),U(46),NA(68),NI(68),NF(68)/4HDELG,4H     ,1ØØ,  Ø,  Ø/
     6 ,U(47),U(48),NA(69),NI(69),NF(69)/4HMERG,4HEL   ,2ØØ,  Ø,  Ø/
     7 ,U(49),U(5Ø),NA(7Ø),NI(7Ø),NF(7Ø)/4HCLAS,4HS    ,  1,  Ø,  Ø/
C
      DATA LRPT/4,5,5,6/
C
C
C ************************* READ USER COMMAND *************************
C
C *** IF NOT STEERING FILE RING BELL AS AUDIBLE PROMPT FOR USER.
C -------------------- UPDATE #7 --------------------
      IF (IBELL .AND. ISLIN .EQ. ISLTTY) CALL TBELL
C ------------------- MWM 3/1Ø/78 -------------------
C
C *** IF REPEAT INDICATOR IS SET, GO READ NEW PARAMETERS.
      IF (REPEAT) GO TO 35
C
C *** PRINT '*' UNLESS LAST COMMAND WAS PLOT OR PLOTNL, OR STEERING
C     FILE IS BEING READ.
      IF (NCMND .EQ. 12 .OR. NCMND .EQ. 13) GO TO 1Ø
5     IF (ISLIN .EQ. ISLTTY) WRITE (ISLOUT,2ØØ)
C
C *** READ USER COMMAND.
1Ø    CALL FREAD (4,1,MODE,CMND)
C
C  WHG 22JAN79
      CALL USRCMD(CMND1,CMND2)
C  WHG 22JAN79
C
```

```
C
C *** LEGAL COMMAND?
        DO 3Ø NCMND=1,NUMCMD
        IF (CMND1 .NE. TF(1,NCMND)) GO TO 3Ø
        IF (CMND2 .EQ. TF(2,NCMND)) GO TO 35
3Ø      CONTINUE
C
C *** NO           PRINT ERROR MESSAGE IF NOT A BLANK COMMAND,
C     AND GO GET NEW COMMAND.
        IF (CMND1 .NE. BLANK) WRITE (ISLOUT,215) CMND
        GO TO 5
C
C ********************** READ COMMAND PARAMETERS **********************
C
C *** TURN REPEAT OFF.
35      REPEAT=.FALSE.
C
C *** RETURN IF NO RECURMENTS ARE REQUIRED.
        IF (NA(NCMND) .NE. Ø) GO TO 4Ø
        IF (NI(NCMND) .NE. Ø) GO TO 4Ø
        IF (NF(NCMND) .EQ. Ø) RETURN
C
C *** COMPUTE REQUIRED NUMBERS OF PARAMETERS:
C       NA1 -- NUMBER OF ALPHANUMERIC PARAMETERS IN FIRST LINE
C       NA2 --    "    "       "           "     IN SECOND LINE.
C       NI1 --    "    " INTEGER PARAMETERS IN FIRST LINE.
C       NI2 --    "    "    "          "    IN SECOND LINE.
C       NI3 --    "    "    "          "    IN THIRD LINE.
C       NI4 --    "    "    "          "    IN FOURTH LINE.
C       NF1 --    "    " FLOATING-POINT PARAMETERS IN FIRST LINE.
C       NF2 --    "    "   "         "         "     IN SECOND LINE.
C       NF3 --    "    "   "         "         "     IN THIRD LINE.
C       NF4 --    "    "   "         "         "     IN FOURTH LINE.
C       NF5 --    "    "   "         "         "     IN FIFTH LINE.
C
4Ø      NA2=NA(NCMND)
        NA1=NA2/1ØØ
        NA2=NA2-1ØØ*NA1
        NI3=NI(NCMND)
        NI1=NI3/1ØØ
        NI2=NI3/1Ø-1Ø*NI1
        NI3=NI3-1Ø*(NI3/1Ø)
        NI4=Ø
        IF (NCMND .EQ. 56) NI4=2
        NF3=NF(NCMND)
        NF1=NF3/1ØØ
        NF2=NF3/1Ø-1Ø*NF1
        NF3=NF3-1Ø*(NF3/1Ø)
        NF4=Ø
        NF5=Ø
C
C *** OBTAIN MODE DIGITS.
```

```
        MODE1=MODE/1Ø
        MODE2=MODE-1Ø*MODE1
C
C *** IF COMMAND IS 'LINEC', SET MAX. NUMBER OF COMPONENTS TO READ.
        IF (NCMND .EQ. 64) NI3=NLCLM
C
C *** IF COMMAND IS 'RGRID' OR 'RGRIDS', AND MODE2 > Ø, NO ACCURACY
C     IS NEEDED.
      IF ((NCMND .EQ. 51 .OR. NCMND .EQ. 52) .AND. MODE2 .GT. Ø) NF3=Ø
C
C *** DETERMINE NUMBER OF F.P. PARAMETERS TO READ FOR 'ELMAT', 'ORMAT',
C     'ETH', 'MSTEEL', 'SSTEEL' AND 'AALLOY'.
        IF (NCMND .EQ. 24) NF3=NTHV
        IF (NCMND .GE. 22 .AND. NCMND .LE. 24) GO TO 5Ø
        IF (NCMND .LT. 36 .OR. NCMND .GT. 38) GO TO 58
        IF (MODE .EQ. Ø) GO TO 55
5Ø      IF (MODE .GT. Ø .AND. MODE .LE. NF3) GO TO 55
52      IERR=1Ø
        RETURN
55      NF3=MODE
C
C *** IF COMMAND IS 'ROTV', AND NO AXIS IS SPECIFIED, THREE ANGLES
C     MUST BE READ.
58      IF (NCMND .NE. 2) GO TO 6Ø
        IF (MODE .LT. Ø .OR. MODE .GT. 3) GO TO 52
        IF (MODE .EQ. Ø) NF3=3
C
C *** UPDATE NUMBER OF F.P. PARAMETERS FOR STANDARD BEAM CROSS-
C     SECTION COMMANDS IF USER WANTS TO SPECIFY AN ATTACHMENT
C     POINT POSITION OR ORIENTATION ANGLE.
6Ø      IF (NCMND .LT. 25 .OR. NCMND .GT. 35) GO TO 65
        IF (NCMND .EQ. 27 .OR. NCMND .EQ. 28) MODE=1Ø*MODE
        MODE1=MODE/1Ø
        MODE2=MODE-1Ø*MODE1
        IF (MODE1 .NE. Ø) NF4=2
        IF (MODE2 .NE. Ø) NF5=1
C
C *** PRINT '>' PROMPT IF NOT STEERING FILE.
65      IF (ISLIN .EQ. ISLTTY) WRITE (ISLOUT,22Ø)
C
C ** READ ALPHANUMERIC PARAMETERS.
        IF (NA1 .EQ. Ø) GO TO 7Ø
        CALL FREAD (3,NA1,L,V)
C
C === IF FIRST ALPHANUMERIC PARAMETER IS BLANK, GO READ NEW COMMAND.
        IF (V1 .EQ. BLANK) GO TO 5
C
7Ø      NA1=NA1+NA1+1
        IF (NA2 .NE. Ø) CALL FREAD (3,NA2,L,V(NA1))
C
C *** READ INTEGER PARAMETERS.
        IF (NI(NCMND) .EQ. Ø) GO TO 75
```

```
C
C === FIRST LINE.
        IF (NI1 .EQ. Ø) GO TO 72
        CALL FREAD (1,NI1,L,V)
C
C --- IF FIRST INTEGER PARAMETER IS NEGATIVE OR ZERO, GO READ NEW
C     COMMAND.
        IF (L1 .LE. Ø) GO TO 5
C
C === SECOND LINE.
72      IF (NI2 .NE. Ø) CALL FREAD (1,NI2,L(NI1+1),V)
C
C --- IS THIS A LINE-GENERATION COMMAND?
        IF (NCMND .LE. 46 .OR. NCMND .GE. 51) GO TO 74
C
C ... YES.  IF MORE THAN ONE LINE SPECIFIED, READ LAST LINE DATA.
        N=LRPT(NCMND-46)
        IF (MODE2 .NE. Ø) N=N+1
        IF (L(N) .GT. Ø) NI3=7
C
C ... IF BEAM ELEMENTS ARE TO BE GENERATED, GEOMETRIC POINTS MUST BE
C     SPECIFIED.
        IF (MODE1 .GE. 1 .AND. ITL .EQ. 2) NI4=2
C
C === IF COMMAND IS 'KPOINT', AND A STRING IS SPECIFIED, COORDINATES
C     OF THE LAST POINT IN THE STRING MUST BE READ.
74      IF (NCMND .EQ. 44 .AND. L2 .GT. L1) NF3=3
C
C === THIRD LINE OF INTEGERS.
        IF (NI3 .NE. Ø) CALL FREAD (1,NI3,L(NI1+NI2+1),V)
C
C === FOURTH LINE.
        IF (NI4 .NE. Ø) CALL FREAD (1,NI4,L(21),V)
C
C *** READ FLOATING-POINT PARAMETERS.
75      IF (NF(NCMND) .EQ. Ø) GO TO 1ØØ
        NA1=NA1+NA2+NA2
        IF (NF1 .EQ. Ø) GO TO 8Ø
        CALL FREAD (2,NF1,L,V(NA1))
        NA1=NA1+NF1
8Ø      IF (NF2 .EQ. Ø) GO TO 85
        CALL FREAD (2,NF2,L,V(NA1))
        NA1=NA1+NF2
85      IF (NF3 .EQ. Ø) GO TO 9Ø
        CALL FREAD (2,NF3,L,V(NA1))
        NA1=NA1+NF3
9Ø      IF (NF4 .EQ. Ø) GO TO 95
        CALL FREAD (2,NF4,L,V(NA1))
        NA1=NA1+NF4
95      IF (NF5 .NE. Ø) CALL FREAD (2,NF5,L,V(NA1))
C
C *** IF NOT STEERING FILE, RING BELL TO INDICATE ALL PARAMETERS READ.
```

```
1ØØ     IF (ISLIN .NE. ISLTTY) RETURN
C ------------------- UPDATE #7 -------------------
       IF (IBELL) CALL TBELL
C ------------------- MWM 3/1Ø/78 -------------------
       RETURN
C
2ØØ    FORMAT (2H *)
215    FORMAT (15H ILL. CMND. -- 2A4)
22Ø    FORMAT (2H >)
       END
```

The following list presents those GIFTS modules and their corresponding command scanner subprograms that were modified to include a call to subroutine USRCMD.

GIFTS Module	Modified Subroutine
BULKLB	COMA
BULKM	COMBM
EDITLB	COMEB
EDITM	COMEM
LOCAL	COML
RESULT	COMR
TRANS	COMT

The next section describes subroutine USRCMD.

7.2 SUBROUTINE USRCMD

Eight commands have been added to the NMFECC CRAY-1 version of GIFTS, mainly to enhance the user's control over graphics output. The following list describes the functions of the new commands.

TEKON turns on the Tektronix screen so that subsequent plotting commands will generate a Tektronix display image. This is the default status of the NMFECC CRAY-1 version of GIFTS.

TEKOFF is the opposite of TEKON. This command is useful if the user is LOGged ON to a non-Tektronix terminal and wishes to plot his model while creating an intermediate plot data file.

STOP is a NMFECC CRAY-1 equivalent to the GIFTS command QUIT. The author of this report sometimes gets confused as to how

to terminate the numerous programs that are run during the course of the day. Therefore, this command and the following one are used to cover just about all the commands that he has ever used to terminate a program.

EXIT is a NMFECC CRAY-1 equivalent to the GIFTS command QUIT.

CMDFL is a command similar in function to the OLB GIFTS command except that any valid file name may be used as a steering file name, instead of the file XJOB.SRC.

TTY: is a NMFECC CRAY-1 equivalent to the GIFTS command END.

TKFON turns on the intermediate plot data file so that subsequent plotting commands will generate a display image in the XJOB.TKF file. The display images accumulate within this file and transcend execution of GIFTS modules; i.e., plots created by BULKM, EDITLB, and RESULT will automatically be appended into the XJOB.TKF file in the same order in which they were generated. The GIFTS UDB is not altered by the TKFON command. No images can be removed from the XJOB.TKF file. If the images within this file are no longer useful, the file should be deleted while at the monitor level.

TKFOFF is the opposite of the TKFON command. TKFOFF is the default status of the NMFECC CRAY-1 version of GIFTS.

The following subroutine listing demonstrates how these commands have been added to GIFTS in an easily updatable manner.

Calling sequence:

```
CALL USRCMD (CMND1,CMND2)
```

Arguments:

CMND1 is the first word of the user's command.

CMND2 is the second word of the user's command.

Language: CFT-FORTRAN

Subroutine listing:

```
      SUBROUTINE USRCMD(CMND1,CMND2)
C *** LAST UPDATED: 22JAN79 WHG
C
C \/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
C \/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
C \/\                                                                  \/\
C \/\ THE ENCLOSED CODE IS DEPENDENT UPON THE COMPUTER SYSTEM BEING     \/\
```

```
C \/\ USED.  THIS IS THE VERSION TO BE USED ON A CRAY-1 COMPUTER.   \/\
C \/\                                                              \/\
      PARAMETER       (NUMCMD = 9),
     1                (NUMCD2 = 18)
C \/\                                                              \/\
C \/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
C \/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
      LOGICAL FILOPN,TKFOPN
C
C \/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
C \/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
C \/\                                                              \/\
C \/\ THE ENCLOSED CODE IS DEPENDENT UPON THE COMPUTER SYSTEM BEING \/\
C \/\ USED.  THIS IS THE VERSION TO BE USED ON A CRAY-1 COMPUTER.   \/\
C \/\                                                              \/\
      DIMENSION       VFL(NUMCD2) ,   TFNEW(2,NUMCMD) ,       XXJOD(2)
C \/\                                                              \/\
C \/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
C \/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
C
      COMMON /CMDSLT/ ISLIN,ISLOUT,ISLTTY
      COMMON /JOB/ XJOB
C
      EQUIVALENCE (TFNEW,VFL)
C
      DATA FILOPN/.FALSE./,TKFOPN/.FALSE./
      DATA EXIT/4HQUIT/, STOP/4HQUIT/
C
C  DATA FOR THE COMMANDS TKFON , TKFOFF , CMDFL , AND TTY:
      DATA
     1   VFL(1),VFL(2)/4HTKFO,4HN   /
     2,  VFL(3),VFL(4)/4HTKFO,4HFF  /
     3,  VFL(5),VFL(6)/4HCMDF,4HL   /
     4,  VFL(7),VFL(8)/4HTTY:,4H    /
     5, VFL(9),VFL(10)/4HEND ,4H    /
     6,VFL(11),VFL(12)/4HSTOP,4H    /
     7,VFL(13),VFL(14)/4HEXIT,4H    /
     8,VFL(15),VFL(16)/4HTEKO,4HN   /
     9,VFL(17),VFL(18)/4HTEKO,4HFF  /
C
C  PROCESS NEW COMMANDS FIRST
      DO 5 I=1,NUMCMD
      IF(CMND1.NE.TFNEW(1,I)) GO TO 5
      IF(CMND2.EQ.TFNEW(2,I)) GO TO 10
5     CONTINUE
C
C  HERE IF NOT ONE OF MY COMMANDS
      RETURN
C
C  IT IS ONE OF MY COMMANDS
10    GO TO (15,16,20,25,30,70,75,80,85) ,I
C
```

```
C  HERE IF TKFON OR TKFOFF
15      CONTINUE
        CALL TKFON
        IF(TKFOPN) GO TO 17
        TKFOPN=.TRUE.
        CALL APNTKF
        XXJOB(1) = XJOB
        XXJOB(2) = 4H TKF
        CALL PERMN (XXJOB,TKFNAME)
        CALL OPNTKF (TKFNAME)
17      CONTINUE
        CMND1=4H
        CMND2=4H
        RETURN
16      CONTINUE
        CALL TKFOFF
        IF(TKFOPN) CALL CLSTKF
        TKFOPN=.FALSE.
        GO TO 17
C
C  HERE IF CMDFL
20      CONTINUE
        FILOPN=.TRUE.
        CALL SLTASN(ISLIN)
        WRITE(ISLTTY,1000)
        READ(ISLTTY,1005) FILNME
        ASSIGN 21 TO IERR
C \/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
C \/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/
C \/\                                                                \/\
C \/\ THE ENCLOSED CODE IS DEPENDENT UPON THE COMPUTER SYSTEM BEING \/\
C \/\ USED.  THIS IS THE VERSION TO BE USED ON A CRAY-1 COMPUTER.    \/\
C \/\                                                                \/\
        CALL OPEN (ISLIN,FILNME,0,LEN,IERR)
C \/\                                                                \/\
C \/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
C \/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/
        IF (LEN .EQ. -1) GO TO 21
        GO TO 17
21      CONTINUE
        WRITE(ISLTTY,1010) FILNME
25      CONTINUE
C
C  HERE IF TTY:
        IF(.NOT.FILOPN) RETURN
C \/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
C \/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/
C \/\                                                                \/\
C \/\ THE ENCLOSED CODE IS DEPENDENT UPON THE COMPUTER SYSTEM BEING \/\
C \/\ USED.  THIS IS THE VERSION TO BE USED ON A CRAY-1 COMPUTER.    \/\
C \/\                                                                \/\
        CALL CLOSE (ISLIN)
```

```
C \/\                                                                    \/\
C \/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
C \/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
      CALL FRESLT(ISLIN)
      ISLIN=ISLTTY
      FILOPN=.FALSE.
      GO TO 17
C
C  HERE IF END.  CHECK FOR COMMAND FILE
3Ø    CONTINUE
      IF(.NOT.FILOPN) RETURN
C
C  IT IS OPEN TRANSFER TO TTY:
      GO TO 25
C
C  STOP COMMAND
7Ø    CMND1=STOP
      RETURN
C
C  EXIT COMMAND
75    CMND1=EXIT
      RETURN
8Ø    CONTINUE
      CALL TEKON
      GO TO 17
85    CONTINUE
      CALL TEKOFF
      GO TO 17
C
1ØØØ  FORMAT('FILENAME ?'/' ')
1ØØ5  FORMAT(A8)
1Ø1Ø  FORMAT('%USRCMD/GIFTS - CANNOT FIND FILE - ',A8)
C
      END
```

Further considerations:

The calls to subroutines TKFON, TKFOFF, APNTKF, OPNTKF, CLSTKF, TEKON, and TEKOFF are satisfied by the TCSTKF software library (see Sect. 1Ø).

7.3 GIFTS MODIFIED STOP SUBROUTINES

In order to be sure that the intermediate plot data files (if any have been created) are closed properly, a call to the TCS subroutine FINITT has been added to the following GIFTS subprograms:

GIFTS Module	Modified Subroutine
BULKLB	STOPA
BULKM	STOPBM
EDITLB	STOPEB
EDITM	STOPEM
LOCAL	STOPL
RESULT	STOPR
TRANS	STOPTR

For example, the underlined statements in the listing of subroutine STOPBM, which follows, demonstrate the necessary modifications to the GIFTS stop subprograms to ensure that graphics is not lost in the intermediate .TKF plot data file.

Modified subroutine STOPBM listing:

```
      SUBROUTINE STOPBM
C *** LAST UPDATED 23JAN79 BY WHG
C *** LAST UPDATED 1/2Ø/78 BY MWM
C *** REWRITTEN MARCH 18, 1976, BY MICHAEL W. MCCABE
C
C     THIS SUBROUTINE FINALIZES THE PROCESSING OF THE CURRENT
C     JOB IN 'BULKM', AND EITHER STOPS (COMMAND 'QUIT') OR
C     RETURNS FOR THE INITIATION OF A NEW JOB (COMMAND 'JOB').
C
      COMMON /CMDSLT/ISLIN,ISLOUT,ISLTTY
      COMMON /COMMND/L(6),LPD(54),V(2Ø),CMND
      COMMON /PAR/LHV(11),LGL(5),NGPT,NGPA,NELT,NKPT,NL,NG,LPX(16),ISTM
      COMMON /PTS/NU,NS,VC(3),SMC(6),LMN,INTP,NBL,NFR,MFP
      COMMON /LIN/XNAME,XNAME2,LLD(2Ø)
      COMMON /GRD/GNAME,GNAME2,LGD(23)
C
      EQUIVALENCE (LLD(2),LINTYP),(LLD(6),NPTL),(LLD(7),NFPL)
      EQUIVALENCE (LPX(4),NDFPM)
      EQUIVALENCE (LGD(2),IGTYPE),(LGD(4),NPTG),(LGD(5),NFPG)
C
      DATA QUIT/4HQUIT/, IGTRAN/4/
C
C
C *** PRINT MESSAGE TO INFORM USER OF WHERE THE TIME ALL WENT.
      WRITE (ISLOUT,1ØØ)
C
C *** CLOSE STEERING FILE, IF ANY.
      IF (ISLIN .NE. ISLTTY) CALL CLOSEF (ISLIN)
C
C *** COMPUTE BASIC FREEDOM PATTERN FOR NODES.
      DO 5 I=1,6
      L(I)=Ø
```

```
        IF (I .LE. NDFPM) L(I)=1
5       LPD(I)=Ø
        CALL PKF (L,MFPS)
C
C *** CREATE POINT DICTIONARY FOR LINE AND GRID INTERNAL NODES
C     AND CHANGE THE LINE AND GRID NODE POINTERS TO USER  NUMBERS.
        NXTUPN=NKPT+1
C
C === LOOP THROUGH LINES.
        IF (NL .EQ. Ø) GO TO 3Ø
        DO 2Ø I=1,NL
        CALL INLIN (I)
C
C --- SKIP COMPOSITE AND ARBITRARY LINES.
        IF (LINTYP .GT. 4) GO TO 2Ø
C
C --- SET LINE'S NODE POINTER TO NEW USER NUMBER OF FIRST
C     INTERNAL NODE.
        NXTSPN=NFPL
        NFPL=NXTUPN
        CALL OUTLIN (I)
C
C --- LOOP THROUGH LINE'S INTERNAL NODES AND SET THEIR USER
C     POINT NUMBERS, POINTERS AND FREEDOM PATTERNS.
        IF (NPTL .EQ. Ø) GO TO 2Ø
        J=NXTUPN+NPTL-1
        IF (J .LE. NGPT) GO TO 8
        NGPT=J
        CALL SIZPTS (NGPT)
8       DO 1Ø J=1,NPTL
        CALL INPTS (NXTSPN)
        NU=NXTUPN
        MFT=MFPS
        CALL OUTPTS (NXTSPN)
        CALL INPTS (NXTUPN)
        NS=NXTSPN
        CALL OUTPTS (NXTUPN)
        NXTSPN=NXTSPN+1
1Ø      NXTUPN=NXTUPN+1
2Ø      CONTINUE
C
C === LOOP THROUGH GRIDS.
3Ø      IF (NG .EQ. Ø) GO TO 6Ø
        DO 5Ø I=1,NG
        CALL INGRD (I)
C
C --- SKIP COMPOSITE AND ARBITRARY GRIDS.
        IF (IGTYPE .GT. IGTRAN) GO TO 5Ø
C
C --- SET GRID'S NODE POINTER TO (NEW) USER NUMBER OF FIRST
C     INTERNAL NODE.
        NXTSPN=NFPG
```

```
        NFPG=NXTUPN
        CALL OUTGRD (I)
C
C --- LOOP THROUGH GRID'S INTERNAL NODES AND SET THEIR USER
C     POINT NUMBERS, POINTERS AND FREEDOM PATTERNS.
        IF (NPTG .EQ. Ø) GO TO 5Ø
        J=NXTUPN+NPTG-1
        IF (J .LE. NGPT) GO TO 35
        NGPT=J
        CALL SIZPTS (NGPT)
35      DO 4Ø J=1,NPTG
        CALL INPTS (NXTSPN)
C -------------------- UPDATE #7 --------------------
        IF (NU .EQ. Ø) GO TO 5Ø
C ------------------- MWM 1/2Ø/78 -------------------
        NU=NXTUPN
        MFP=MFPS
        CALL OUTPTS (NXTSPN)
        CALL INPTS (NXTUPN)
        NS=NXTSPN
        CALL OUTPTS (NXTUPN)
        NXTSPN=NXTSPN+1
4Ø      NXTUPN=NXTUPN+1
5Ø      CONTINUE
C
C *** LOOP THROUGH KEY POINTS, AND SET THEIR FREEDOM PATTERN.
6Ø      DO 7Ø I=1,NKPT
        NP=I
        CALL INPTU (NP)
        IF (NP .EQ. Ø) GO TO 7Ø
        MFP=MFPS
        CALL OUTPTS (NP)
7Ø      CONTINUE
C
C *** NOW SEARCH THROUGH NODES FOR ANY NOT GENERATED BY 'BULKM'
C     (THEY WILL HAVE USER NUMBERS SET, BUT NO FREEDOMS), AND
C     GIVE THEM NEW USER NUMBERS AFTER THE LAST GRID.
        IF (NXTUPN .GT. NGPT) GO TO 9Ø
        DO 8Ø NXTSPN=1,NGPT
        CALL INPTS (NXTSPN)
        IF (NU .EQ. Ø) GO TO 8Ø
        IF (MFP .NE. Ø) GO TO 8Ø
        NU=NXTUPN
        MFP=MFPS
        CALL OUTPTS (NXTSPN)
        CALL INPTS (NXTUPN)
        NS=NXTSPN
        CALL OUTPTS (NXTUPN)
        NXTUPN=NXTUPN+1
8Ø      CONTINUE
C
C *** CLOSE POINT, ELEMENT, LINE AND GRID FILES.
```

```
9Ø      CALL CLSPTS
        CALL CLSELT
        CALL CLSLIN
        CALL CLSGRD
C
C *** UPDATE AND OUTPUT THE JOB'S PARAMETER FILE.
        ISTM=1
        NCPA-NCPT
        CALL OUTPRP
C
C *** STOP IF COMMAND WAS 'QUIT', ELSE RETURN.
C
C  23JAN79 WHG
        CALL TERASE
        CALL FINITT(Ø,78Ø)
        CALL THOME
C  23JAN79 WHG
C
        IF (CMND .EQ. QUIT) STOP
        RETURN
1ØØ     FORMAT (/23H ASSIGNING USER NUMBERS)
        END
```

7.4 LOGICAL FUNCTION NEWPG

The function of logical function NEWPG is to query the user about whether to continue the listing display of information that is about to be written off the screen of a Tektronix terminal. In the NMFECC CRAY-1 version of GIFTS this question is never asked if the COSMOS[13] controller is executing the job.

Two of several reasons why users should execute GIFTS under the control of COSMOS are (1) better utilization of the computer resources for large problems and (2) production of archival copies of the information contained within a GIFTS UDB via the listing of required information into the user's COSMOS LOG file. A modification was made to subroutine NEWPG to facilitate the latter process. It is assumed that the user wishes to see all the information; therefore, the interactive prompt (which is produced after every 3Ø lines of typeout) is suppressed if anything but the user's terminal is controlling the job.

Calling sequence:

IF (NEWPG(Ø)) do something

Arguments: none

Further explanation:

In the above example the executable FORTRAN instruction represented by 'do something' could be a read statement.

Language: CFT-FORTRAN

Subroutine listing:

```
      LOGICAL FUNCTION NEWPG (I)
C ***   UPDATED: 21JAN79 WHG
C *** UPDATED 3/4/76 BY MWM
C *** WRITTEN JULY 11, 1975, BY MICHAEL W. MCCABE
C
C     THIS IS A LOGICAL FUNCTION TO BE CALLED WHENEVER A SCREEN FULL
C     OF INFORMATION HAS BEEN OUTPUT TO THE TERMINAL, AND IT IS
C     DESIRED TO WAIT FOR THE USER TO ASK FOR CONTINUATION OR TERMIN-
C     ATION.  'NEWPG' IS A LOGICAL VARIABLE THAT RETURNS THE STATUS:
C       .TRUE.  - USER REQUESTED NEXT PAGE
C       .FALSE. - USER REQUESTED TERMINATION
C
      LOGICAL BATCHJ
C
      COMMON /CMDSLT/ISLIN,ISLOUT,ISLTTY
C
      NEWPG=.TRUE.
C
C  WHG 21JAN79
C
C              IF IT IS A BATCH JOB DO NOT ASK THE QUESTION.
C
      IF(BATCHJ(Ø)) RETURN
C  WHG 21JAN79
C
C
C *** PRINT CONTINUATION MESSAGE AND GET RESPONSE.
      WRITE (ISLTTY,1ØØ)
      READ (ISLTTY,1Ø5) ANS
C
C *** SET VALUE OF 'NEWPG'.
      IF (ANS .EQ. 1RS) NEWPG=.FALSE.
      RETURN
C
1ØØ   FORMAT (48H TO CONTINUE, STRIKE 'RETURN'; TO STOP, TYPE 'S')
1Ø5   FORMAT (R1)
      END
```

8. NMFECC CRAY-1 GIFTS SUPPORTING SOFTWARE

This section discusses three miscellaneous subprograms required by the NMFECC CRAY-1 version of GIFTS.

8.1 SUBROUTINE DC8T4

The function of subroutine DC8T4 is to reformat an ASCII text string from an A8 format into a 2A4 format suitable for the subprograms of GIFTS.

Calling sequence:

CALL DC8T4 (A8,T3A4)

Arguments:

A8	is the input ASCII text string formatted with A8.
T3A4	is the returned ASCII text string formatted with 2A4.

Language: CFT-FORTRAN

Subroutine listing:

```
C
        SUBROUTINE DC8T4 (IN,OUT)
C
C \/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
C \/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
C \/\                                                               \/\
C \/\ THE ENCLOSED CODE IS DEPENDENT UPON THE COMPUTER SYSTEM BEING \/\
C \/\ USED.  THIS IS THE VERSION TO BE USED ON A CRAY-1 COMPUTER.   \/\
C \/\                                                               \/\
C
        DIMENSION OUT(1)
C
C
C  CONVERT FROM A8 TO 2A4
        DO 100 I=1,2
        ICHRPOS=(I-1)*4
        CALL ZMOVECHR (OUT(I),0,IN,ICHRPOS,4)
100     CONTINUE
C \/\                                                               \/\
C \/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
C \/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
        RETURN
        END
```

8.2 LOGICAL FUNCTION BATCHJ

The function of logical function BATCHJ is to determine whether the batch controller COSMOS is running the job. If COSMOS is controlling the job or, in fact, if anything but the user's terminal is controlling the job, then BATCHJ will be true.

Calling sequence:

IF (BATCHJ(0)) do something

Arguments: none

Further explanation:

In the above example the executable FORTRAN instruction represented by 'do something' could be a read statement. In fact, statements of this type have been placed into the NMFECC CRAY-1 version of GIFTS so that the interactive questions asked by the GIFTS information commands will not be asked if a GIFTS module is being executed with COSMOS as its controller. The GIFTS subroutine NEWPG was modified to eliminate these unnecessary questions.

Language: CFT-FORTRAN

Subroutine listing:

```
C
        LOGICAL FUNCTION BATCHJ (I)
C
C \/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
C \/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
C \/\                                                               \/\
C \/\ THE ENCLOSED CODE IS DEPENDENT UPON THE COMPUTER SYSTEM BEING \/\
C \/\ USED.  THIS IS THE VERSION TO BE USED ON A CRAY-1 COMPUTER.   \/\
C \/\                                                               \/\
C
        DIMENSION I(2)
C
        BATCHJ=.FALSE.
C
C  FIND OUT WHO IS CONTROLLING THE JOB.
        IF (IZM04 (I) .NE. 0) 999,100
100     CONTINUE
C
C  IF IT'S COSMOS, I(2) WILL BE = 6HCOSMOS
        IF (I(2) .NE. 0) BATCHJ=.TRUE.
999     CONTINUE
C \/\                                                               \/\
C \/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
C \/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
```

```
      RETURN
      END
```

8.3 LOGICAL FUNCTION PRESNT

Determining the presence or absence of a file in the user's disk area is the function of logical function PRESNT.

Calling sequence:

IF (PRESNT(XEXT)) do something

Arguments:

XEXT	is the file extension. The file name (job name) is assumed to be located in the first word of common /JOB/. There can be a maximum of four characters in the job name and a maximum of three characters in the file extension. The file extension begins at the second-character position in XEXT.

Language: CFT-FORTRAN

Subroutine listing:

```
C
      LOGICAL FUNCTION PRESNT (IEXT)
C
      DIMENSION JJOB(2)
C
      COMMON  /JOB/   IJOB
C
      JJOB(1)=IJOB
      JJOB(2)=IEXT
      CALL PERMN (JJOB,FNAME)
      PRESNT=.TRUE.
C
C \/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
C \/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
C \/\                                                              \/\
C \/\ THE ENCLOSED CODE IS DEPENDENT UPON THE COMPUTER SYSTEM BEING \/\
C \/\ USED.  THIS IS THE VERSION TO BE USED ON A CRAY-1 COMPUTER.   \/\
C \/\                                                              \/\
      LEN=0
      ITYPE=-1
      CALL FREEUS (ISLOT)
      CALL OPEN (ISLOT,FNAME,ITYPE,LEN)
      IF (LEN .EQ. -1) PRESNT=.FALSE.
      CALL CLOSE (ISLOT)
C \/\                                                              \/\
C \/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
```

```
C \/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
        RETURN
        END
```

9. THE TCS SOFTWARE LIBRARY FOR THE NMFECC CRAY-1

The TCS software as furnished by Tektronix requires six subroutines to be written for it to execute properly on specific computer hardware. The remainder of the TCS software is written in a subset of FORTRAN that is generally machine independent. The machine-dependent portion of the TCS software library was written for the NMFECC CRAY-1 computer and is described below.

9.1 SUBROUTINE ADEIN

Subroutine ADEIN performs to Tektronix Terminal Control System specifications and may be used to input right-justified, zero-filled ASCII characters from the user's terminal.

Calling sequence:

```
CALL ADEIN (NCHAR,KADE)
```

Argument list:

NCHAR is the number of characters to be translated (unmodified by this subroutine).

KADE is the right-justified ASCII character array stored one character per word (unmodified by this subroutine).

Language: CFT-FORTRAN

Subprogram listing:

```
C
      SUBROUTINE ADEIN(NCHAR,IARAY)
      DIMENSION IARAY(1),KARAY(72),MSG(10)
      DO 10 I=1,72
      KARAY(I)=32
10    CONTINUE
C
C \/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
C \/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
C \/\                                                                        \/\
C \/\ THE ENCLOSED CODE IS DEPENDENT UPON THE COMPUTER SYSTEM BEING           \/\
C \/\ USED.  THIS IS THE VERSION TO BE USED ON A CRAY-1 COMPUTER.            \/\
C \/\                                                                        \/\
      IF (IZGETMR (MSG,72,0,IADE) .NE. 0) 20,30
C \/\                                                                        \/\
C \/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
C \/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
30    CONTINUE
```

```
      NCHAR=IADE
C
C \/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
C \/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
C \/\                                                               \/\
C \/\ THE ENCLOSED CODE IS DEPENDENT UPON THE COMPUTER SYSTEM BEING \/\
C \/\ USED.  THIS IS THE VERSION TO BE USED ON A CRAY-1 COMPUTER.   \/\
C \/\                                                               \/\
      CALL ZRJCHARZ (KARAY,MSG,IADE)
C \/\                                                               \/\
C \/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
C \/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
      DO 40 I=1,NCHAR
      IARAY(I)=KARAY(I)
40    CONTINUE
      RETURN
20    CONTINUE
      NCHAR=0
      RETURN
      END
```

9.2 SUBROUTINE ADEOUT

Subroutine ADEOUT performs to Tektronix Terminal Control System specifications and may be used to output right-justified, zero-filled ASCII characters to the user's terminal.

Calling sequence:

CALL ADEOUT (NCHAR,KADE)

Argument list:

NCHAR	is the number of characters to be translated (unmodified by this subroutine).
KADE	is the right-justified ASCII character array stored one character per word (unmodified by this subroutine).

Language: CFT-FORTRAN

Subprogram listing:

```
C
      SUBROUTINE ADEOUT (NADE,KADE)
C
      DIMENSION KADE(NADE),MSG(10)
C
C
C \/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
C \/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
C \/\                                                               \/\
C \/\ THE ENCLOSED CODE IS DEPENDENT UPON THE COMPUTER SYSTEM BEING \/\
```

```
C \/\ USED.  THIS IS THE VERSION TO BE USED ON A CRAY-1 COMPUTER.   \/\
C \/\                                                              \/\
      CALL ZPAKCHRZ (MSG,KADE,NADE)
      IF (IZSNDMR (MSG,NADE,0) .NE. 0) 10,20
C \/\                                                              \/\
C \/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
C \/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
10    CONTINUE
20    CONTINUE
      RETURN
      END
```

9.3 SUBROUTINE KA12AS

Subroutine KA12AS performs to Tektronix Terminal Control System specifications and may be used to convert left-justified ASCII characters (stored one character per word) into right-justified, zero-filled characters (stored one character per word).

Calling sequence:

```
CALL KA12AS (NCHAR,KA1,KADE)
```

Argument list:

NCHAR — is the number of characters to be translated (unmodified by this subroutine).

KA1 — is the left-justified ASCII character array stored one character per word (unmodified by this subroutine).

KADE — is the right-justified, zero-filled ASCII character array stored one character per word (returned by this subroutine).

Language: CFT-FORTRAN

Subprogram listing:

```
C
      SUBROUTINE KA12AS (NADE,KA1,KADE)
C
      DIMENSION KA1(1) , KADE(1)
C
      DO 100 I=1,NADE
C
C \/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
C \/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
C \/\                                                              \/\
C \/\ THE ENCLOSED CODE IS DEPENDENT UPON THE COMPUTER SYSTEM BEING \/\
C \/\ USED.  THIS IS THE VERSION TO BE USED ON A CRAY-1 COMPUTER.   \/\
```

```
C \/\                                                                  \/\
      CALL ZRJCHARZ (KADE(I),KA1(I),1)
C \/\                                                                  \/\
C \/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
C \/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
100   CONTINUE
      RETURN
      END
```

9.4 SUBROUTINE KAM2AS

Subroutine KAM2AS performs to Tektronix Terminal Control System specifications and may be used to convert left-justified ASCII characters (stored five characters per word) into right-justified, zero-filled characters (stored one character per word).

Calling sequence:

```
CALL KAM2AS (NCHAR,KAM,KADE)
```

Argument list:

NCHAR is the number of characters to be translated (unmodified by this subroutine).

KAM is the left-justified ASCII character array stored five characters per word (unmodified by this subroutine).

KADE is the right-justified, zero-filled ASCII character array stored one character per word (returned by this subroutine).

Language: CFT-FORTRAN

Subprogram listing:

```
C
      SUBROUTINE KAM2AS (NADE,KAM,KADE)
C
      DIMENSION KAM(1) , KADE(1)
C
C
C \/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
C \/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
C \/\                                                                  \/\
C \/\ THE ENCLOSED CODE IS DEPENDENT UPON THE COMPUTER SYSTEM BEING     \/\
C \/\ USED.  THIS IS THE VERSION TO BE USED ON A CRAY-1 COMPUTER.       \/\
C \/\                                                                  \/\
      CALL ZRJCHARZ (KADE,KAM,NADE)
C \/\                                                                  \/\
C \/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
```

```
C \/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
        RETURN
        END
```

9.5 SUBROUTINE KAS2A1

Subroutine KAS2A1 performs to Tektronix Terminal Control System specifications and may be used to convert right-justified ASCII characters (stored one zero-filled character per word) into left-justified, blank-filled characters (stored one character per word).

Calling sequence:

CALL KAS2A1 (NCHAR,KADE,KA1)

Argument list:

NCHAR	is the number of characters to be translated (unmodified by this subroutine).
KADE	is the right-justified, zero-filled ASCII character array stored one character per word (unmodified by this subroutine).
KA1	is the left-justified ASCII character array stored one character per word (returned by this subroutine).

Language: CFT-FORTRAN

Subprogram listing:

```
C
        SUBROUTINE KAS2A1(NCHAR,KADE,KA1)
C
        DIMENSION KADE(1) , KA1(1)
C
        DO 100 I=1,NCHAR
C
C \/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
C \/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
C \/\                                                              \/\
C \/\ THE ENCLOSED CODE IS DEPENDENT UPON THE COMPUTER SYSTEM BEING \/\
C \/\ USED.  THIS IS THE VERSION TO BE USED ON A CRAY-1 COMPUTER.   \/\
C \/\                                                              \/\
        CALL ZPAKCHRZ (KA1(I),KADE(I),1)
C \/\                                                              \/\
C \/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
C \/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
100     CONTINUE
        RETURN
        END
```

9.6 SUBROUTINE KAS2AM

Subroutine KAS2AM performs to Tektronix Terminal Control System specifications and may be used to convert right-justified ASCII characters (stored one zero-filled character per word) into left-justified, blank-filled characters (stored one character per word).

Calling sequence:

CALL KAS2AM (NCHAR,KADE,KAM)

Argument list:

NCHAR	is the number of characters to be translated (unmodified by this subroutine).
KADE	is the right-justified ASCII character array stored one character per word (unmodified by this subroutine).
KAM	is the left-justified, blank-filled ASCII character array stored five characters per word (returned by this subroutine).

Language: CFT-FORTRAN

Subprogram listing:

```
C
      SUBROUTINE KAS2AM (NCHAR,KADE,KAM)
C
      DIMENSION KADE(1) , KAM(1)
C
C
C \/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
C \/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
C \/\                                                               \/\
C \/\ THE ENCLOSED CODE IS DEPENDENT UPON THE COMPUTER SYSTEM BEING \/\
C \/\ USED.  THIS IS THE VERSION TO BE USED ON A CRAY-1 COMPUTER.   \/\
C \/\                                                               \/\
      CALL ZPAKCHRZ (KAM,KADE,NCHAR)
C \/\                                                               \/\
C \/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
C \/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
      RETURN
      END
```

9.7 SUBROUTINE SLP

Subroutine SLP may be used to put a program to sleep for a specified number of seconds. This subroutine has been substituted for subroutine IOWAIT in the TCS subroutines HDCOPY, NEWPAG, and ERASE. The reason for this substitution was discussed in the system considerations section of Ref. 7.

Calling sequence:

```
     CALL SLP (ISEC)
```

Argument list:

ISEC is the number of seconds to sleep.

Language: CFT-FORTRAN

Subprogram listing:

```
C
      SUBROUTINE SLP(ISEC)
C
C  CALCULATE THE INTEGER NUMBER OF MICROSECONDS TO SLEEP.
      IMICROSE=1000000*ISEC
C
C \/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
C \/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
C \/\                                                                        \/\
C \/\ THE ENCLOSED CODE IS DEPENDENT UPON THE COMPUTER SYSTEM BEING           \/\
C \/\ USED.  THIS IS THE VERSION TO BE USED ON A CRAY-1 COMPUTER.            \/\
C \/\                                                                        \/\
C
C  SUSPEND THE PROGRAM EXECUTION FOR THE SPECIFIED
C  NUMBER OF MICROSECONDS.
      IF (IZSUSPND(IMICROSE).NE.0) 100,100
C \/\                                                                        \/\
C \/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
C \/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
100   CONTINUE
      RETURN
      END
```

10. THE GFTTKF SOFTWARE LIBRARY

The following subprograms list the modified versions of the TCSTKF software library used by the ORNL version of GIFTS. This software was written to emulate the DECsystem-10 version of the TCSTKF software library.[5]

The following is a brief description of the function, calling sequence, and argument list for the subprograms available in the NMFECC CRAY-1 TCSTKF software library.

10.1 SUBROUTINE ADEOUT

Subprogram ADEOUT is a replacement for the similarly named subroutine in the TCS software library. The function of subprogram ADEOUT is to transmit ASCII characters to the user's terminal and/or to the user's .TKF disk file.

Calling sequence:

CALL ADEOUT (IADE,JADE)

Arguments:

IADE	is the number of ASCII Decimal Equivalent (ADE) characters to be transmitted to the terminal and/or the .TKF file.
JADE	is the array containing the right-justified ASCII characters.

Language: CFT-FORTRAN

Subroutine listing:

```
      SUBROUTINE ADEOUT (NADE,KADE)
C
C \/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
C \/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
C \/\                                                               \/\
C \/\ THE ENCLOSED CODE IS DEPENDENT UPON THE COMPUTER SYSTEM BEING \/\
C \/\ USED.  THIS IS THE VERSION TO BE USED ON A CRAY-1 COMPUTER.   \/\
C \/\                                                               \/\
C
C  WRITTEN ON 01AUG79 BY WHG
C
      LOGICAL BATCHJ
C
      LOGICAL TEK , TKF
C
```

```
      INTEGER TKFADE , WRDTKFRC , DPOINT , POINT , CHRPT , TKFIOC
C
C  DIMENSION STATEMENT FOR ADE CHARACTERS PASSED THROUGH ARGUMENT LIST
C  AND MESSAGE ARRAY FOR SENDING CHARACTERS TO TEKTRONIX
C
      DIMENSION       KADE(NADE) , MSG(2Ø) , TMPIOA(9)
C
      COMMON  /TKFCOMBK/      TEK , TKF , TKFIOC , WRDTKFRC ,
     1                        DPOINT , POINT , CHRPT , TKFIOA(9) ,
     2                        NAMETKF , MAXTKFRC , ZAPTKF
C
      COMMON  /TKFADEBF/      TKFADE(1)
C
      COMMON  /TKFIOCBF/      IOCTKFBF(1)
C
      DATA    EXTRABIT /Ø/ , TMPNAME /8H%ADE TMP/
C
C  ARE WE SENDING ADE CHARACTERS TO THE TKF FILE
C
      IF ( .NOT. TKF) GO TO 5ØØ
      IF (TKFIOC .EQ. Ø) GO TO 5ØØ
C
C  HERE IF WE ARE DOING THE TKF OPTION.  BEGIN BY INITIALIZING
C  THE ADE CHARACTER POINTER.
C
      IADE = Ø
1ØØ   CONTINUE
C
C  INCREMENT POINTER INTO ADE ARRAY
C
      IADE = IADE + 1
C
C  ARE WE FINISHED LOOPING THROUGH THE ADE ARRAY
C
      IF (IADE .GT. NADE) GO TO 5ØØ
C
C  TRANSFER THE CURRENT ADE CHARACTER FROM ADE TO THE TKFADE ARRAY
C
      CALL ZMOVEBIT (TKFADE,POINT,KADE(IADE),57,7)
C
C  INCREMENT THE Ø-ORIGIN ARRAY BIT POINTER INTO TKFADE
C
      POINT = POINT + 7
C
C  INCREMENT THE CHARACTERS TRANSFERRED SUM
C
      CHRPT = CHRPT + 1
C
C  IF A MULTIPLE OF 5 ADE CHARACTERS HAS BEEN TRANSFERRED, INSERT A
C  SINGLE EXTRA BIT INTO TKFADE IN ORDER THAT BINARY TRANSMISSION OF THIS
C  FILE BACK TO A DECSYSTEM-1Ø WILL PRODUCE AN ASCII LEFT-
C  JUSTIFIED FILE.
```

```
C
      IF (CHRPT .NE. 5) GO TO 1Ø5
      CHRPT = Ø
C
C  HERE TO INSERT THE EXTRA BIT
C
      CALL ZMOVEBIT (TKFADE,POINT,EXTRABIT,63,1)
C
C  INCREMENT THE Ø-ORIGIN ARRAY BIT POINTER INTO TKFADE
C  TO REFLECT INSERTION OF THE EXTRA BIT.
C
      POINT = POINT + 1
C
1Ø5   CONTINUE
C
C  CHECK TO SEE IF THE TKFADE ARRAY IS FULL.  IF NOT, GO
C  BACK TO STATEMENT 1ØØ AND CONTINUE PROCESSING ADE CHARACTERS.
C
      IF ((POINT / 64) + 1 .LE. WRDTKFRC) GO TO 1ØØ
C
C  HAS AN I/O REQUEST BEEN ISSUED
C
      IF (DPOINT) 115 , 115 , 125
C
C  YES, SINCE DPOINT HAS BEEN INCREMENTED.  SEE IF THE I/O REQUEST
C  IS STILL PENDING.
C
125   CONTINUE
      IF (IZIOSTTZ(TKFIOC,Ø,ICOUNT,TKFIOA)) 11Ø , 115 , 92Ø
C
C  HERE IF I/O REQUEST IS STILL PENDING.  GIVE UP CPU
C
11Ø   CONTINUE
      IF (IZIOSTTZ(TKFIOC,1,ICOUNT,TKFIOA)) 115 , 115 , 92Ø
C
C  HERE IF I/O REQUEST HAS BEEN FULFILLED
C
115   CONTINUE
C
C  TRANSFER TKFADE TO IOCTKFBF
C
      CALL ZMOVEWRD (IOCTKFBF,TKFADE,WRDTKFRC,DUMMY)
C
C  ISSUE AN I/O REQUEST
C
      IF (IZIDKOUT(TKFIOC,IOCTKFBF,DPOINT,WRDTKFRC,TKFIOA)
     1                  .NE. Ø ) 95Ø , 12Ø
C
C  HERE IF NO ERRORS DURING I/O REQUEST
C
12Ø   CONTINUE
C
```

```
C   INCREMENT THE Ø-ORIGIN DISK WORD ADDRESS FOR THE TKF FILE
C
        DPOINT = DPOINT + WRDTKFRC
C
C   CHECK TO SEE IF THE FILE HAS ROOM FOR THE NEXT I/O
C
        IF (DPOINT / WRDTKFRC .EQ. MAXTKFRC) GO TO 3ØØ
C
C   HERE IF THERE IS ROOM.
C   RESET THE Ø-ORIGIN BIT POINTER INTO ARRAY TKFADE AND THE TKFADE ARRAY.
C
13Ø     CONTINUE
        POINT = POINT - WRDTKFRC * 64
        CALL ZVXMITS (TKFADE,Ø,WRDTKFRC)
C
C   NOW TRANSFER THE REMAINING BITS IN THE LAST WORD OF THE TKFADE
C   ARRAY TO THE FIRST WORD OF THE TKFADE ARRAY
C
        CALL ZMOVEBIT (TKFADE,Ø,TKFADE(WRDTKFRC + 1),Ø,POINT)
        TKFADE(WRDTKFRC + 1) = Ø
C
C   GO COMPLETE LOOP OVER ADE CHARACTERS
C
        GO TO 1ØØ
3ØØ     CONTINUE
C
C   HERE IF THE FILE NEEDS TO BE EXPANDED
C
C
C   IF FILE IS ALREADY FULL BUMP UP MAXIMUM SIZE
C
        MAXTKFRC = MAXTKFRC + MAXTKFRC
C
C   GET ANOTHER IOC AND CREATE A NEW LARGER FILE TO TRANSFER
C   THE ADE CHARACTERS INTO.  ALSO, DESTROY THE TEMPORARY FILE, IF
C   IT WAS PRESENT FROM AN ABORTED RUN.
C
        IF (IZDSTROY(TMPNAME,Ø) .NE. Ø) 3Ø4 , 3Ø4
3Ø4     CONTINUE
        CALL FREEIOC (TMPIOC)
        IF (TMPIOC) 923 , 923 , 3Ø5
3Ø5     CONTINUE
        LENBIG = MAXTKFRC * WRDTKFRC
        IF (IZCREATE(TMPIOC,TMPNAME,LENBIG) .NE. Ø) 925 , 31Ø
C
C   NOW WE NEED TO TRANSFER THE ADE CHARACTERS FROM
C   XJOB.TKF TO THE TEMPORARY FILE WITHOUT POINTER UPDATE
C
31Ø     CONTINUE
C
C   GIVE UP CPU UNTIL I/O IS FINISHED.  THEN CLEAR TRANSFER POINTER
C
```

```
      IF (IZIOSTTZ(TKFIOC,1,ICOUNT,TKFIOA)) 315 , 315 , 94Ø
315   CONTINUE
      IDPOINT = Ø
C
C  NOW LOOP THROUGH THE NUMBER OF FULL RECORDS
C
33Ø   CONTINUE
C
C  READ FROM XJOB.TKF THE APPROPRIATE RECORD
C
      IF (IZIDKIN(TKFIOC,IOCTKFBF,IDPOINT,WRDTKFRC,TKFIOA)
     1                .NE. Ø) 93Ø , 34Ø
C
C  GIVE UP CPU UNTIL THIS I/O IS COMPLETE
C
34Ø   CONTINUE
      IF (IZIOSTTZ(TKFIOC,1,ICOUNT,TKFIOA)) 35Ø , 35Ø , 94Ø
C
C  HAVE WE ISSUED AN I/O REQUEST FOR THE TEMPORARY FILE
C
35Ø   CONTINUE
      IF (IDPOINT) 36Ø , 36Ø , 37Ø
C
C  YES, SINCE IDPOINT IS GREATER THAN ZERO, IT HAS BEEN
C  INCREMENTED.  SEE IF REQUEST IS STILL PENDING FOR
C  THE TEMPORARY FILE
C
37Ø   CONTINUE
      IF (IZIOSTTZ(TMPIOC,Ø,IC,TMPIOA)) 38Ø , 36Ø , 95Ø
C
C  HERE IF I/O REQUEST IS STILL PENDING FOR TEMPORARY
C  FILE. GIVE UP CPU UNTIL FINISHED.
C
38Ø   CONTINUE
      IF (IZIOSTTZ(TMPIOC,1,IC,TMPIOA)) 36Ø , 36Ø , 95Ø
C
C  HERE IF I/O REQUEST FOR TEMPORARY FILE IS FINISHED
C
36Ø   CONTINUE
C
C  TRANSFER IOCTKFBF TO TKFADE
C
      CALL ZMOVEWRD (TKFADE,IOCTKFBF,WRDTKFRC,DUMMY)
C
C  WRITE OUT TO THE TEMPORARY FILE
C
      ISPOINT = IDPOINT
      IF (IZIDKOUT(TMPIOC,TKFADE,ISPOINT,WRDTKFRC,TMPIOA)
     1                .NE. Ø) 96Ø , 39Ø
C
C  HERE IF NO ERRORS DURING OUTPUT REQUEST.  INCREMENT IDPOINT.
C
```

```
39Ø     CONTINUE
        IDPOINT = IDPOINT + WRDTKFRC
C
C  END OF LOOP TO TRANSFER FULL RECORDS
C
        IF (IDPOINT .NE. DPOINT) GO TO 33Ø
C
C  GIVE UP CPU UNTIL LAST I/O IS COMPLETE
C
        IF (IZIOSTTZ(TMPIOC,1,IC,TMPIOA)) 395 , 395 , 95Ø
395     CONTINUE
C
C  FINISH FILE BOOKKEEPING.  CLOSE AND DESTROY XJOB.TKF
C
        IF (IZCLOSE(TKFIOC,3) .NE. Ø) 981 , 42Ø
42Ø     CONTINUE
        IF (IZDSTROY(NAMETKF,Ø) .NE. Ø) 982 , 43Ø
C
C  CLOSE AND CHANGE NAME OF THE TEMPORARY FILE TO XJOB.TKF
C
43Ø     CONTINUE
        IF (IZCLOSE(TMPIOC,3) .NE.Ø) 983 , 44Ø
44Ø     CONTINUE
        IF (IZCHANGE(TMPNAME,NAMETKF,Ø,Ø) .NE.Ø) 984 , 45Ø
C
C  NOW REOPEN THE EXPANDED FILE
C
45Ø     CONTINUE
        CALL FREEIOC (TKFIOC)
        IF (IZOPEN(TKFIOC,NAMETKF,LENBIG) .NE. Ø) 985 , 13Ø
C
C  FINALLY WE'RE DONE WITH THE EXPANSION
C
C
C  HERE FOR TEK PROCESSING
C
5ØØ     CONTINUE
C
C  ARE WE SENDINC ADE CHARACTERS TO THE TEKTRONIX
C
        IF ( .NOT. TEK) GO TO 8ØØ
C
C  YES, BUT DO NOT SEND THE CHARACTERS TO A BATCH JOB
C
        IF (BATCHJ(Ø)) GO TO 8ØØ
C
C  PACK THE RIGHT-JUSTIFIED, ZERO-FILLED CHARACTERS IN KADE INTO
C  THE LEFT-JUSTIFIED ASCII STRING MSG
C
        CALL ZPAKCHRZ (MSG,KADE,NADE)
C
C  SEND THE CHARACTERS TO THE TERMINAL
```

```
C
        IF (IZSNDMR(MSG,NADE,Ø) .EQ. Ø) 51Ø , 51Ø
51Ø     CONTINUE
C
C  NORMAL RETURN FROM THIS SUBROUTINE
C
8ØØ     CONTINUE
C
        RETURN
C
C  ERROR BRANCHES
C
92Ø     CONTINUE
C
C  ERROR ISSUING I/O REQUEST INQUIRY
C
        WRITE(59,1Ø2Ø)
        GO TO 999
923     CONTINUE
C
C  NO IOC'S AVAILABLE
C
        WRITE(59,1Ø23)
        GO TO 999
925     CONTINUE
C
C  CANNOT OPEN THE TEMPORARY FILE
C
        WRITE(59,1Ø25)
        GO TO 999
93Ø     CONTINUE
C
C  ERROR READING XJOB.TKF
C
        WRITE(59,1Ø3Ø)
        GO TO 999
94Ø     CONTINUE
C
C  ERROR EXAMINING READ STATUS
C
        WRITE(59,1Ø4Ø)
        GO TO 999
945     CONTINUE
C
C  ERROR EXAMINING WRITE STATUS
C
        WRITE(59,1Ø45)
        GO TO 999
96Ø     CONTINUE
C
C  ERROR WRITING TEMPORARY FILE
C
```

```
        WRITE(59,1Ø6Ø)
        GO TO 999
95Ø     CONTINUE
C
C  ERROR DURING I/O REQUEST
C
        WRITE(59,1Ø5Ø)
C
C  ERROR RETURN FROM THIS SUBROUTINE
C
        GO TO 999
981     CONTINUE
C
C  ERROR CLOSING FILE
C
        WRITE(59,1Ø81)
        GO TO 999
982     CONTINUE
C
C  ERROR DESTROYING FILE
C
        WRITE(59,1Ø82)
        GO TO 999
983     CONTINUE
C
C  ERROR CLOSING FILE
C
        WRITE(59,1Ø83)
        GO TO 999
984     CONTINUE
C
C  ERROR CHANGING NAME
C
        WRITE(59,1Ø84)
        GO TO 999
985     CONTINUE
C
C  ERROR REOPENING FILE
C
        WRITE(59,1Ø85)
        GO TO 999
999     CONTINUE
        STOP
C
1Ø2Ø    FORMAT('?ADEOUT/GIFTS - ERROR 92Ø. FATAL'/
     1          'TKF I/O STATUS REQUEST FAILED')
C
1Ø23    FORMAT('?ADEOUT/GIFTS - ERROR 923 - FATAL'/
     1  'SYSTEM ERROR - NO IOC AVAILABLE')
C
1Ø25    FORMAT('?ADEOUT/GIFTS - ERROR 925 - FATAL'/
     1  'CANNOT CREATE THE TEMPORARY FILE %ADE TMP')
```

```
C
1Ø3Ø    FORMAT('?ADEOUT/GIFTS - ERROR 93Ø - FATAL'/
     1  'CANNOT READ XJOB.TKF')
C
1Ø4Ø    FORMAT('?ADEOUT/GIFTS - ERROR 94Ø - FATAL'/
     1  'I/O STATUS REQUEST FAILED')
C
1Ø45    FORMAT('?ADEOUT/GIFTS - ERROR 945 - FATAL'/
     1  'I/O STATUS REQUEST FAILED')
C
1Ø5Ø    FORMAT('?ADEOUT/GIFTS - ERROR 95Ø. FATAL'/
     1          'TKF I/O REQUEST FAILED')
C
1Ø6Ø    FORMAT('?ADEOUT/GIFTS - ERROR 96Ø - FATAL'/
     1  'CANNOT WRITE TEMPORARY TKF FILE')
C
1Ø81    FORMAT('?ADEOUT/GIFTS - ERROR 981 - FATAL'/
     1  'CANNOT CLOSE XJOB.TKF')
C
1Ø82    FORMAT('?ADEOUT/GIFTS - ERROR 982 - FATAL'/
     1  'CANNOT DESTROY XJOB.TKF')
C
1Ø83    FORMAT('?ADEOUT/GIFTS - ERROR 983 - FATAL'/
     1  'CANNOT CLOSE TEMPORARY TKF FILE')
C
1Ø84    FORMAT('?ADEOUT/GIFTS - ERROR 984 - FATAL'/
     1  'CANNOT CHANGE THE NAME OF THE TEMPORARY FILE')
C
1Ø85    FORMAT('?ADEOUT/GIFTS - ERROR 985 - FATAL'/
     1  'CANNOT REOPEN XJOB.TKF FILE')
C \/\                                                                    \/\
C \/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
C \/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
C
        END
```

1Ø.2 SUBROUTINE APNTKF

A switch is set by calling subprogram APNTKF that causes subsequent ADE characters to be appended to an existing .TKF file.

Calling sequence:

```
CALL APNTKF
```

Arguments: none

Language: CFT-FORTRAN

Subroutine listing:

```
C
      SUBROUTINE APNTKF
C
      LOGICAL TEK , TKF , ZAPTKF
C
      COMMON  /TKFCOMBK/      TEK , TKF , TKFIOC , WRDTKFRC ,
     1                        DPOINT , POINT , CHRPT , TKFIOA(9) ,
     2                        NAMETKF , MAXTKFRC , ZAPTKF
C
C  TURN ON THE APPEND SWITCH
C
      ZAPTKF = .TRUE.
C
      RETURN
      END
```

10.3 ENTRY POINT CLSTKF

Subprogram CLSTKF closes and releases the .TKF file. It is an entry point in subprogram FINITT (see the next section).

Calling sequence:

```
    CALL CLSTKF
```

Arguments: none

10.4 SUBROUTINE FINITT

Subprogram FINITT is a replacement for the similarly named subroutine in the TCS software library. The function of subprogram FINITT is to close and release the .TKF file.

Calling sequence:

```
    CALL FINITT (IX,IY)
```

Arguments: ignored

Language: CFT-FORTRAN

Subroutine listing:

```
      SUBROUTINE FINITT (IX,IY)
C
C \/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
C \/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
C \/\                                                                \/\
C \/\ THE ENCLOSED CODE IS DEPENDENT UPON THE COMPUTER SYSTEM BEING \/\
C \/\ USED.  THIS IS THE VERSION TO BE USED ON A CRAY-1 COMPUTER.   \/\
```

```
C \/\                                                                \/\
C
C  WRITTEN ON Ø6AUG79 BY WHG
C
      LOGICAL TEK , TKF
C
      INTEGER TKFADE , WRDTKFRC , DPOINT , POINT , CHRPT , TKFIOC
      INTEGER WRDLASRC
C
      COMMON  /TKFCOMBK/      TEK , TKF , TKFIOC , WRDTKFRC ,
     1                        DPOINT , POINT , CHRPT , TKFIOA(9) ,
     2                        NAMETKF , MAXTKFRC , ZAPTKF
C
      COMMON  /TKFADEBF/      TKFADE(1)
C
C  MOVE THE ALPHANUMERIC CURSOR TO THE SPECIFIED COORDINATES
C
      CALL MOVABS (IX,IY)
      CALL ANMODE
      CALL TSEND
C
C  DEFINE ENTRY POINT CLSTKF
C
      ENTRY CLSTKF
C
C  ARE WE DOING A TKF FILE
C
      IF ( .NOT. TKF) GO TO 8ØØ
      IF (TKFIOC .EQ. Ø) GO TO 8ØØ
C
C  HERE IF WE'RE DOING TKF'S.  CHECK TO SEE IF THERE IS AN I/O PENDING.
C
      IF (IZIOSTTZ(TKFIOC,1,ICOUNT,TKFIOA)) 1ØØ , 1ØØ , 9ØØ
C
C  CALCULATE THE FILE SIZE AND OUTPUT THE LAST RECORD
C
1ØØ   CONTINUE
      WRDLASRC = (POINT/64) + 1
      LEN = DPOINT + WRDLASRC
      IF (IZIDKOUT(TKFIOC,TKFADE,DPOINT,WRDLASRC,TKFIOA)
     1                .NE. Ø) 91Ø , 12Ø
12Ø   CONTINUE
      IF (IZIOSTTZ(TKFIOC,1,ICOUNT,TKFIOA)) 13Ø , 13Ø , 9ØØ
C
C  CLOSE AND SHRINK FILE
C
13Ø   CONTINUE
      IF (IZCLOSE(TKFIOC,3) .NE. Ø) 92Ø , 14Ø
14Ø   CONTINUE
C
C  SPECIAL CASE CHECK
C
```

```
        IF (LEN .EQ. MAXTKFRC * WRDTKFRC) GO TO 8ØØ
        IF (IZSHRINK(NAMETKF,LEN,Ø) .NE. Ø) 93Ø , 8ØØ
8ØØ     CONTINUE
C
C  NORMAL ENDING TO THIS SUBROUTINE
C
        TKFIOC = Ø
        RETURN
9ØØ     CONTINUE
C
C  ERROR ISSUING I/O REQUEST INQUIRY
C
        WRITE(59,1ØØØ)
        GO TO 999
91Ø     CONTINUE
C
C  ERROR DURING I/O REQUEST
C
        WRITE(59,1Ø1Ø)
        GO TO 999
92Ø     CONTINUE
C
C  ERROR CLOSING FILE
C
        WRITE(59,1Ø2Ø)
        GO TO 999
93Ø     CONTINUE
C
C  ERROR SHRINKING FILE
C
        WRITE(59,1Ø3Ø)
        GO TO 8ØØ
C
C  ERROR RETURN FROM THIS SUBROUTINE
C
999     CONTINUE
        STOP
C
1ØØØ    FORMAT('?FINITT/GIFTS - ERROR 9ØØ. FATAL'/
     1          'TKF I/O STATUS REQUEST FAILED')
C
1Ø1Ø    FORMAT('?FINITT/GIFTS - ERROR 91Ø. FATAL'/
     1          'TKF I/O REQUEST FAILED')
C
C
1Ø2Ø    FORMAT('?FINITT/GIFTS - ERROR 92Ø - FATAL'/
     1  'CANNOT CLOSE TEMPORARY TKF FILE')
C
1Ø3Ø    FORMAT('%FINITT/GIFTS - ERROR 93Ø - WARNING'/
     1  'CANNOT SHRINK THE TKF FILE')
C \/\                                                                  \/\
C \/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
```

```
C \/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
      END
```

10.5 SUBROUTINE OPNTKF

The function of subprogram OPNTKF is to open the XJOB.TKF file in the user's disk area and to initialize the GFTTKF plotting software.

Calling sequence:

IF (.NOT. OPNTKF(DUMMY)) do something

Arguments: none

Language: CFT-FORTRAN

Subroutine listing:

```
      SUBROUTINE OPNTKF (NAGRTKF)
C
C \/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/
C \/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/
C \/\                                                                 \/\
C \/\ THE ENCLOSED CODE IS DEPENDENT UPON THE COMPUTER SYSTEM BEING \/\
C \/\ USED.  THIS IS THE VERSION TO BE USED ON A CRAY-1 COMPUTER.     \/\
C \/\                                                                 \/\
C
C  WRITTEN ON 01AUG79 BY WHG
C
C  DEFINE THE MAXIMUM NUMBER OF RECORDS ALLOWABLE IN A
C  TKF FILE (MAXREC), THE NUMBER OF WORDS IN A TKF RECORD (NWORDS), AND
C  THE NUMBER OF WORDS PLUS 1 IN A TKF RECORD (NWORDS1)
C
      PARAMETER        (MAXREC = 100) ,
     1                 (NWORDS = 512) ,
     2                 (NWORDS1 = 513)
C
      LOGICAL TEK , TKF , ZAPTKF
C
      INTEGER TKFADE , WRDTKFRC , DPOINT , POINT , CHRPT , TKFIOC
C
C  DIMENSION STATEMENT FOR TEMPORARY FILE IOAREA
C
      DIMENSION        TMPIOA(9)
C
      COMMON  /TKFCOMBK/      TEK , TKF , TKFIOC , WRDTKFRC ,
     1                        DPOINT , POINT , CHRPT , TKFIOA(9) ,
     2                        NAMETKF , MAXTKFRC , ZAPTKF
C
      COMMON  /TKFADEBF/      TKFADE(NWORDS1)
C
      COMMON  /TKFIOCBF/      IOCTKFBF(NWORDS)
C
```

```
      DATA     TMPNAME /8H%TKF_TMP/ , ZAPTKF /.FALSE./
      DATA     TEK /.TRUE./ , TKF/.FALSE./
C
C  HAS THE FILE ALREADY BEEN OPENED.  IF SO, DO NOT DO IT AGAIN.
C
      IF (TKFIOC .NE. Ø) GO TO 8ØØ
C
C  TRANSFER TKF FILE NAME TO TKFCOMBK
C
      NAMETKF = NAGRTKF
C
C  DO NOT ASSUME SUCCESS FROM THIS ROUTINE
C
      TKF = .FALSE.
C
C  TRANSFER THE SIZING INFORMATION TO THE COMMON BLOCK /TKFCOMBK/
C
      MAXTKFRC = MAXREC
      WRDTKFRC = NWORDS
C
C  HERE IF THE TKF FILE NEEDS TO BE OPENED OR INITIALIZED. FIRST GET
C  AN IOC
C
      CALL FREEIOC (TMPIOC)
C
C  DID WE GET ONE?
C
      IF (TMPIOC .EQ. Ø) GO TO 9ØØ
C
C  DOES THE FILE EXIST IN THE USER'S DISK AREA
C
      IF (IZOPEN(TMPIOC,NAMETKF,LEN,IRWE) .NE. Ø) 1ØØ , 5ØØ
C
C  HERE IF TKF FILE DOES NOT EXIST.  FIRST CLEAR ERROR.
C
1ØØ   CONTINUE
      IF (IZCLOSE(TMPIOC,3) .NE. Ø) 1Ø5 , 1Ø5
C
C  CALCULATE THE TKF FILE LENGTH
C
1Ø5   CONTINUE
      LEN = MAXTKFRC * WRDTKFRC
C
C  INITIALIZE TKFIOC
C
      CALL FREEIOC (TKFIOC)
C
C  CREATE THE TKF FILE
C
      IF (IZCREATE(TKFIOC,NAMETKF,LEN) .NE. Ø) 91Ø , 11Ø
C
C  INITIALIZE (1) THE Ø-ORIGIN BIT ADDRESS INTO TKFADE (POINT), (2) THE
```

```
C  Ø-ORIGIN DISK WORD ADDRESS INTO THE TKF FILE (DPOINT), AND (3) THE TKF
C  CHARACTER TRANSFER SUM (CHRPT).
C
11Ø     CONTINUE
        POINT = Ø
        DPOINT = Ø
        CHRPT = Ø
        TKF = .TRUE.
C
C  END OF CREATION SECTION OF SUBROUTINE OPNTKF.
C
        GO TO 8ØØ
C
C  HERE IF TKF FILE ALREADY EXISTS, ARE WE APPENDING
C
5ØØ     CONTINUE
        IF (.NOT. ZAPTKF) GO TO 1ØØ
C
C  HERE IF WE ARE APPENDING, HOW MANY FULL RECORDS ARE THERE
C
        ITR = LEN / WRDTKFRC
C
C  IF FILE IS ALREADY FULL BUMP UP MAXIMUM SIZE
C
        IF (ITR .GE. MAXTKFRC) MAXTKFRC = ITR + MAXREC
C
C  GET ANOTHER IOC AND CREATE A NEW LARGER FILE TO TRANSFER
C  THE ADE CHARACTERS INTO.  ALSO, DESTROY THE TEMPORARY FILE, IF
C  IT WAS PRESENT FROM AN ABORTED RUN.
C
        IF (IZDSTROY(TMPNAME,Ø) .NE. Ø) 5Ø4 , 5Ø4
5Ø4     CONTINUE
        CALL FREEIOC (TKFIOC)
        IF (TKFIOC) 92Ø ; 92Ø , 5Ø5
5Ø5     CONTINUE
        LENBIG = MAXTKFRC * WRDTKFRC
        IF (IZCREATE(TKFIOC,TMPNAME,LENBIG) .NE. Ø) 925 , 51Ø
C
C  NOW WE NEED TO TRANSFER THE ADE CHARACTERS FROM
C  XJOB.TKF TO THE TEMPORARY FILE WHILE UPDATING THE APPRO-
C  PRIATE POINTERS
C
51Ø     CONTINUE
        POINT = Ø
        DPOINT = Ø
        CHRPT = Ø
C
C  NOW LOOP THROUGH THE NUMBER OF FULL RECORDS
C
        IF (ITR .EQ. Ø) GO TO 52Ø
C
        DO 53Ø I = 1 , ITR
```

```
C
C  READ FROM XJOB.TKF THE APPROPRIATE RECORD
C
      IF (IZIDKIN(TMPIOC,IOCTKFBF,DPOINT,WRDTKFRC,TMPIOA)
     1                .NE. Ø) 93Ø , 54Ø
C
C  GIVE UP CPU UNTIL THIS I/O IS COMPLETE
C
54Ø   CONTINUE
      IF (IZIOSTTZ(TMPIOC,1,ICOUNT,TMPIOA)) 55Ø , 55Ø , 94Ø
C
C  HAVE WE ISSUED AN I/O REQUEST FOR THE TEMPORARY FILE
C
55Ø   CONTINUE
      IF (DPOINT) 56Ø , 56Ø , 57Ø
C
C  YES, SINCE DPOINT IS CREATER THAN ZERO, IT HAS BEEN
C  INCREMENTED.  SEE IF REQUEST IS STILL PENDING FOR
C  THE TEMPORARY FILE
C
57Ø   CONTINUE
      IF (IZIOSTTZ(TKFIOC,Ø,IC,TKFIOA)) 58Ø , 56Ø , 95Ø
C
C  HERE IF I/O REQUEST IS STILL PENDING FOR TEMPORARY
C  FILE. GIVE UP CPU UNTIL FINISHED.
C
58Ø   CONTINUE
      IF (IZIOSTTZ(TKFIOC,1,IC,TKFIOA)) 56Ø , 56Ø , 95Ø
C
C  HERE IF I/O REQUEST FOR TEMPORARY FILE IS FINISHED
C
56Ø   CONTINUE
C
C  TRANSFER IOCTKFBF TO TKFADE
C
      CALL ZMOVEWRD (TKFADE,IOCTKFBF,WRDTKFRC,DUMMY)
C
C  WRITE OUT TO THE TEMPORARY FILE
C
      ISPOINT = DPOINT
      IF (IZIDKOUT(TKFIOC,TKFADE,ISPOINT,WRDTKFRC,TKFIOA)
     1                .NE. Ø) 96Ø , 59Ø
C
C  HERE IF NO ERRORS DURING OUTPUT REQUEST.  INCREMENT DPOINT.
C
59Ø   CONTINUE
      DPOINT = DPOINT + WRDTKFRC
C
C  END OF LOOP TO TRANSFER FULL RECORDS
C
53Ø   CONTINUE
C
```

```
C  GIVE UP CPU UNTIL LAST I/O IS COMPLETE
C
      IF (IZIOSTTZ(TKFIOC,1,IC,TKFIOA)) 595 , 595 , 95Ø
595   CONTINUE
C
C  NOW TAKE CARE OF THE LAST (PARTIALLY FULL) RECORD
C  HOW MANY WORDS ARE LEFT
C
52Ø   CONTINUE
      LEFT = LEN - DPOINT
C
C  SPECIAL CASE.  IF LEFT IS Ø, THEN LAST WORD IN PREVIOUS
C  RECORD WAS PARTIALLY WRITTEN.  DO NOT READ ANOTHER
C  RECORD
C
      IF (LEFT .EQ. Ø) GO TO 62Ø
C
C  ZERO OUT TKFADE
C
      CALL ZVXMITS (TKFADE,Ø,WRDTKFRC + 1)
C
C  READ THE APPROPRIATE RECORD FROM XJOB.TKF AND GIVE UP CPU
C  UNTIL FINISHED WITH READ
C
      IF (IZIDKIN(TMPIOC,TKFADE,DPOINT,LEFT,TMPIOA)
     1                  .NE. Ø) 97Ø , 61Ø
61Ø   CONTINUE
      IF (IZIOSTTZ(TMPIOC,1,ICOUNT,TMPIOA)) 62Ø , 62Ø , 98Ø
C
C  NOW UPDATE APPROPRIATE POINTERS
C
62Ø   CONTINUE
C
C  ADJUST DPOINT FOR SPECIAL CASE
C
      IF (LEFT .EQ. Ø) DPOINT = DPOINT - WRDTKFRC
      POINT = ((LEN * 64) / 36) * 36 - DPOINT * 64
      CHRPT = Ø
C
C  FINISH FILE BOOKKEEPING.  CLOSE AND DESTROY XJOB.TKF
C
      IF (IZCLOSE(TMPIOC,3) .NE. Ø) 981 , 72Ø
72Ø   CONTINUE
      IF (IZDSTROY(NAMETKF,Ø) .NE. Ø) 982 , 73Ø
C
C  CLOSE AND CHANGE NAME OF THE TEMPORARY FILE TO XJOB.TKF
C
73Ø   CONTINUE
      IF (IZIOSTTZ(TKFIOC,1,IC,TKFIOA)) 735 , 735 , 95Ø
735   CONTINUE
      IF (IZCLOSE(TKFIOC,3) .NE.Ø) 983 , 74Ø
74Ø   CONTINUE
```

```
        IF (IZCHANGE(TMPNAME,NAMETKF,Ø,Ø) .NE.Ø) 984 , 75Ø
C
C  NOW REOPEN THE EXPANDED FILE
C
75Ø     CONTINUE
        CALL FREEIOC (TKFIOC)
        IF (IZOPEN(TKFIOC,NAMETKF,LENBIG) .NE. Ø) 985 , 76Ø
C
C  FINALLY WE'RE DONE
C
76Ø     CONTINUE
        TKF = .TRUE.
8ØØ     CONTINUE
        RETURN
C
C  ERROR BRANCHES
C
9ØØ     CONTINUE
C
C  NO IOC'S AVAILABLE
C
        WRITE(59,1ØØØ)
        GO TO 999
91Ø     CONTINUE
C
C  TKF FILE CREATE FAILED
C
        WRITE(59,1Ø1Ø)
C
C  CLOSE FILE, RELEASE IOC, AND RETURN
C
        IF (IZCLOSE(TKFIOC,Ø) .NE. Ø) 915 , 915
915     CONTINUE
        TKFIOC = Ø
        GO TO 8ØØ
92Ø     CONTINUE
C
C  NO IOC'S AVAILABLE
C
        WRITE(59,1Ø2Ø)
        GO TO 999
925     CONTINUE
C
C  CANNOT CREATE TEMPORARY FILE
C
        WRITE(59,1Ø25)
        GO TO 999
93Ø     CONTINUE
C
C  ERROR READING XJOB.TKF
C
        WRITE(59,1Ø3Ø)
```

```
        GO TO 999
94Ø     CONTINUE
C
C  ERROR EXAMINING READ STATUS
C
        WRITE(59,1Ø4Ø)
        GO TO 999
95Ø     CONTINUE
C
C  ERROR EXAMINING WRITE STATUS
C
        WRITE(59,1Ø5Ø)
        GO TO 999
96Ø     CONTINUE
C
C  ERROR WRITING TEMPORARY FILE
C
        WRITE(59,1Ø6Ø)
        GO TO 999
97Ø     CONTINUE
C
C  ERROR READING XJOB.TKF
C
        WRITE(59,1Ø7Ø)
        GO TO 999
98Ø     CONTINUE
C
C  ERROR EXAMINING READ STATUS
C
        WRITE(59,1Ø8Ø)
        GO TO 999
981     CONTINUE
C
C  ERROR CLOSING FILE
C
        WRITE(59,1Ø81)
        GO TO 999
982     CONTINUE
C
C  ERROR DESTROYING FILE
C
        WRITE(59,1Ø82)
        GO TO 999
983     CONTINUE
C
C  ERROR CLOSING FILE
C
        WRITE(59,1Ø83)
        GO TO 999
984     CONTINUE
C
C  ERROR CHANGING NAME
```

```
C
        WRITE(59,1Ø84)
        GO TO 999
985     CONTINUE
C
C  ERROR REOPENING FILE
C
        WRITE(59,1Ø85)
        GO TO 999
C
C  ERROR TERMINATION FROM THIS ROUTINE
C
999     CONTINUE
        STOP
C
1ØØØ    FORMAT('?OPNTKF/GIFTS - ERROR 9ØØ - FATAL'/
     1  'SYSTEM ERROR - NO IOC AVAILABLE')
C
1Ø1Ø    FORMAT('%OPNTKF/GIFTS - ERROR 91Ø - WARNING'/
     2  'CANNOT CREATE TKF FILE.  TRY AGAIN LATER')
C
1Ø2Ø    FORMAT('?OPNTKF/GIFTS - ERROR 92Ø - FATAL'/
     1  'SYSTEM ERROR - NO IOC AVAILABLE')
C
1Ø25    FORMAT('?OPNTKF/GIFTS - ERROR 925 - FATAL'/
     1  'CANNOT CREATE THE TEMPORARY FILE %TKF TMP')
C
1Ø3Ø    FORMAT('?OPNTKF/GIFTS - ERROR 93Ø - FATAL'/
     1  'CANNOT READ XJOB.TKF')
C
1Ø4Ø    FORMAT('?OPNTKF/GIFTS - ERROR 94Ø - FATAL'/
     1  'I/O STATUS REQUEST FAILED')
C
1Ø5Ø    FORMAT('?OPNTKF/GIFTS - ERROR 95Ø - FATAL'/
     1  'I/O STATUS REQUEST FAILED')
C
1Ø6Ø    FORMAT('?OPNTKF/GIFTS - ERROR 96Ø - FATAL'/
     1  'CANNOT WRITE TEMPORARY TKF FILE')
C
1Ø7Ø    FORMAT('?OPNTKF/GIFTS - ERROR 97Ø - FATAL'/
     1  'CANNOT READ XJOB.TKF')
C
1Ø8Ø    FORMAT('?OPNTKF/GIFTS - ERROR 98Ø - FATAL'/
     1  'I/O STATUS REQUEST FAILED')
C
1Ø81    FORMAT('?OPNTKF/GIFTS - ERROR 981 - FATAL'/
     1  'CANNOT CLOSE XJOB.TKF')
C
1Ø82    FORMAT('?OPNTKF/GIFTS - ERROR 982 - FATAL'/
     1  'CANNOT DESTROY XJOB.TKF')
C
1Ø83    FORMAT('?OPNTKF/GIFTS - ERROR 983 - FATAL'/
```

```
     1  'CANNOT CLOSE TEMPORARY TKF FILE')
C
1Ø84    FORMAT('?OPNTKF/GIFTS - ERROR 984 - FATAL'/
     1  'CANNOT CHANGE THE NAME OF THE TEMPORARY FILE')
C
1Ø85    FORMAT('?OPNTKF/GIFTS - ERROR 985 - FATAL'/
     1  'CANNOT REOPEN XJOB.TKF FILE')
C
C \/\                                                                    \/\
C \/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
C \/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
        END
```

1Ø.6 SUBROUTINE TEKOFF

The function of subprogram TEKOFF is to send the current TCS internal buffer to subroutine ADEOUT and then set a switch to prevent future plotting on the user's Tektronix terminal.

Calling sequence:

```
     CALL TEKOFF
```

Arguments: none

Language: CFT-FORTRAN

Subroutine listing:

```
        SUBROUTINE TEKOFF
C
C \/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
C \/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
C \/\                                                                    \/\
C \/\ THE ENCLOSED CODE IS DEPENDENT UPON THE COMPUTER SYSTEM BEING       \/\
C \/\ USED.  THIS IS THE VERSION TO BE USED ON A CRAY-1 COMPUTER.        \/\
C \/\                                                                    \/\
C
C  WRITTEN ON Ø1AUG79 BY WHG
C
        LOGICAL TEK , TKF
C
        COMMON  /TKFCOMBK/      TEK , TKF
C
C  FIRST CHECK TO SEE IF THERE ARE ANY ADE CHARACTERS LEFT IN THE
C  TCS OUTPUT BUFFER.  IF THERE ARE, SEND THEM TO THE TERMINAL.
C
        IF (LEFTIO(Ø) .NE. Ø) CALL TSEND
C
C  TURN THE TEK SWITCH OFF
C
        TEK = .FALSE.
```

```
C
      RETURN
C \/\                                                                  \/\
C \/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
C \/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
      END
```

10.7 SUBROUTINE TEKON

The function of subprogram TEKON is to send the current TCS internal buffer to subroutine ADEOUT and then set a switch to enable future plotting on the user's Tektronix terminal.

Calling sequence:

```
     CALL TEKON
```

Arguments: none

Language: CFT-FORTRAN

Subroutine listing:

```
      SUBROUTINE TEKON
C
C \/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
C \/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
C \/\                                                                  \/\
C \/\ THE ENCLOSED CODE IS DEPENDENT UPON THE COMPUTER SYSTEM BEING     \/\
C \/\ USED.  THIS IS THE VERSION TO BE USED ON A CRAY-1 COMPUTER.       \/\
C \/\                                                                  \/\
C
C  WRITTEN ON Ø1AUG79 BY WHG
C
      LOGICAL TEK , TKF , BATCHJ
C
      COMMON  /TKFCOMBK/      TEK , TKF
C
C  FIRST CHECK TO SEE IF THERE ARE ANY ADE CHARACTERS LEFT IN THE
C  TCS OUTPUT BUFFER.  IF THERE ARE, SEND THEM TO THE TERMINAL.
C
      IF (LEFTIO(Ø) .NE. Ø) CALL TSEND
C
C  TURN THE TEK SWITCH ON
C
      TEK = .TRUE.
C
C  IS THIS PROGRAM BEING EXECUTED FROM A BATCH JOB.  IF IT IS
C  DON'T CLUTTER THE BATCH LOG FILE WITH ADE CHARACTERS.
C
      IF (BATCHJ(Ø)) TEK = .FALSE.
C
```

```
      RETURN
C \/\                                                                  \/\
C \/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
C \/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
      END
```

10.8 SUBROUTINE TKFOFF

The function of subprogram TKFOFF is to send the current TCS internal buffer to subroutine ADEOUT and then set a switch to prevent future plotting on the user's .TKF file.

Calling sequence:

```
CALL TKFOFF
```

Arguments: none

Language: CFT-FORTRAN

Subroutine listing:

```
      SUBROUTINE TKFOFF
C
C \/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
C \/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
C \/\                                                                  \/\
C \/\ THE ENCLOSED CODE IS DEPENDENT UPON THE COMPUTER SYSTEM BEING     \/\
C \/\ USED.  THIS IS THE VERSION TO BE USED ON A CRAY-1 COMPUTER.       \/\
C \/\                                                                  \/\
C
C  WRITTEN ON Ø1AUG79 BY WHG
C
      LOGICAL TEK , TKF
C
      COMMON  /TKFCOMBK/      TEK , TKF
C
C  FIRST CHECK TO SEE IF THERE ARE ANY ADE CHARACTERS LEFT IN THE
C  TCS OUTPUT BUFFER.  IF THERE ARE, SEND THEM TO THE TERMINAL.
C
      IF (LEFTIO(Ø) .NE. Ø) CALL TSEND
C
C  TURN THE TKF SWITCH OFF
C
      TKF = .FALSE.
C
      RETURN
C \/\                                                                  \/\
C \/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
C \/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
      END
```

10.9 SUBROUTINE TKFON

The function of subprogram TKFON is to send the current TCS internal buffer to subroutine ADEOUT and then set a switch to enable future plotting on the user's .TKF file.

Calling sequence:

CALL TKFON

Arguments: none

Language: CFT-FORTRAN

Subroutine listing:

```
      SUBROUTINE TKFON
C
C \/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
C \/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
C \/\                                                               \/\
C \/\ THE ENCLOSED CODE IS DEPENDENT UPON THE COMPUTER SYSTEM BEING \/\
C \/\ USED.  THIS IS THE VERSION TO BE USED ON A CRAY-1 COMPUTER.  \/\
C \/\                                                               \/\
C
C  WRITTEN ON 01AUG79 BY WHG
C
      LOGICAL TEK , TKF
C
      INTEGER TKFIOC
C
      COMMON  /TKFCOMBK/      TEK , TKF , TKFIOC
C
C  FIRST CHECK TO SEE IF THERE ARE ANY ADE CHARACTERS LEFT IN THE
C  TCS OUTPUT BUFFER.  IF THERE ARE, SEND THEM TO THE TERMINAL.
C
      IF (LEFTIO(0) .NE. 0) CALL TSEND
C
C  TURN THE TKF 3WITCH ON
C
      TKF = .TRUE.
C
C  MAKE SURE THAT WE HAVE A TKF FILE TO DO OUTPUT TO
C
      IF (TKFIOC .LE. 0) TKF = .FALSE.
C
      RETURN
C \/\                                                               \/\
C \/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
C \/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
      END
```

11. CONCLUSIONS

The system software written for the NMFECC CRAY-1 version of GIFTS has been presented. It is believed that this software represents an efficient implementation of GIFTS on a CRAY-1. Furthermore, a technique for maintaining an enhanced command instruction set has been developed and can be maintained with a minimum of effort. These added commands permit generation of report-quality graphics directly while using the GIFTS plotting capability.

THIS PAGE

WAS INTENTIONALLY

LEFT BLANK

REFERENCES

1. H. A. Kamel and M. W. McCabe, "The GIFTS System: Version 4B Systems Manual," University of Arizona, Tucson, Arizona (July 1978).
2. W. H. Gray, System Software for the ORNL Version of GIFTS 4B, Oak Ridge National Laboratory Report ORNL/TM-7009, Oak Ridge, Tennessee (November 1979).
3. "FORTLIB, Computer Documentation," National Magnetic Fusion Energy Computer Center, Lawrence Livermore Laboratory, Livermore, California (May 1979).
4. "BASELIB, Computer Documentation," National Magnetic Fusion Energy Computer Center, Lawrence Livermore Laboratory, Livermore, California (May 1979).
5. W. H. Gray, "A User's Guide to the TCSTKF Software Library," document in preparation.
6. "Tektronix PLOT 1Ø Terminal Control System User Manual," Tektronix, Inc., Beaverton, Oregon (February 1977).
7. W. H. Gray, User and System Considerations for the TCSTEK Software Library, Oak Ridge National Laboratory Report ORNL/TM-6918, Oak Ridge, Tennessee (August 1979).
8. R. D. Burris and W. H. Gray, "A Generalized Plotting Facility," Proc. Digital Equipment Computer Users Society, San Francisco, California (November 1978).
9. "NETOUT, Computer Documentation," National Magnetic Fusion Energy Computer Center, Lawrence Livermore Laboratory, Livermore, California (May 1979).
1Ø. W. H. Gray, A User's Guide to Program GFTMOV: A Computer Program to Access and Format a GIFTS 4B UDB into a Data Base for the MOVIE.BYU Program, Oak Ridge National Laboratory Report ORNL/TM-7021, Oak Ridge, Tennessee (January 1980).
11. W. H. Gray, "Program GFTDMP," undocumented.
12. W. H. Gray, "Program GFTSAP," undocumented.
13. "COSMOS, Computer Documentation," National Magnetic Fusion Energy Computer Center, Lawrence Livermore Laboratory, Livermore, California (December 1979).

THIS PAGE

WAS INTENTIONALLY

LEFT BLANK

ORNL/TM-7369
Dist. Category UC-32

INTERNAL DISTRIBUTION

1. G. A. Aramayo
2-6. T. V. Baudry
7. W. D. Cain
8. W. G. Dodge
9-13. W. H. Gray
14. S. K. Iskander
15. R. E. Textor
16-17. Laboratory Records Dept.
18-19. Central Research Library
20. Document Reference Section
21. Laboratory Records, ORNL-RC
22. ORNL Patent Office
23-24. Fusion Energy Division Library
25. Fusion Energy Division Reports Office

EXTERNAL DISTRIBUTION

26. D. J. Anthony, General Electric Company, Building 23, Room 290, 1 River Road, Schenectady, NY 12345
27. J. L. Armand, Institut de Recherches de la Construction Navale, 47 Rue de Monceau, 75008, Paris, France
28. J. Barger, General Electric Company-AEG, Building 500, Mail Stop K190, Cincinnati, OH 45215
29. R. Bell, Department of Mechanical and Aerospace Engineering, Carlton University, Ottawa, Ontario, K1S 5B6, Canada
30. J. Bernard, Department of Mechanical Engineering, Michigan State University, East Lansing, MI 48824
31. M. Binder, Voest-Alpine International Corporation, Lincoln Building, 60 East 42nd Street, New York, NY 10017
32. D. Blackfield, Plasma Fusion Center, Massachusetts Institute of Technology, 167 Albany Street, Cambridge, MA 02139
33. M. Brightenstein, United Computing Systems, 2525 Washington, Kansas City, MO 64108
34. R. W. Conn, Department of Chemical, Nuclear, and Thermal Engineering, University of California, Los Angeles, CA 90024
35. J. G. Craig, U.S. Department of Energy, Morgantown Energy Technology Center, P.O. Box 880, Collins Ferry Road, Morgantown, WV 26505
36. H. J. Davis, Harry Diamond Laboratories, 2800 Powder Mill Road, Adelphi, MD 20783
37. J. A. Deaker, Computer Engineering Applications, Limited, 9th Floor, Amoco Building, 201 Pacific Highway, North Sydney, Australia
38. S. O. Dean, Science Applications, Inc., 2 Professional Drive, Suite 249, Gaithersburg, MD 20760
39. D. Desfosse, Grumman Data Systems Corporation, Calverton Production Center, Mail Stop B29-07, Calverton, NY 11933
40. J. Donaghey, Data General Corporation, 15 Turnpike Road, Mail Stop 92200, Westborough, MA 01581
41. H. K. Forsen, Exxon Nuclear Company, 777 106th Avenue, NE, Bellevue, WA 98004

42. H. P. Furth, Princeton Plasma Physics Laboratory, P.O. Box 451, Princeton, NJ 08540
43. S. E. Gabrielse, Westinghouse Electric Corporation, Research and Development Center, 1310 Beulah Road, Pittsburgh, PA 15235
44. J. J. Gallagher, Raytheon Company, Missile Systems Division, Mail Stop C-20, Bedford, MA 01730
45. V. J. Garg, Association of American Railroads, 3140 South Federal Street, Chicago, IL 60616
46. R. W. Gould, California Institute of Technology, Building 116-81, Pasadena, CA 91109
47. W. Hayes, Beth Israel Hospital, Department of Orthopedic Surgery, 330 Brookline Avenue, Boston, MA 02215
48. G. Helliwell, Computer Aided Process Group, Defense Research Centre Salisbury, P.O. Box 2151, Adelaide, South Australia 5001
49. R. L. Hirsch, Exxon Research and Engineering Company, P.O. Box 101, Florham Park, NJ 07932
50. K. Hunten, General Dynamics, Fort Worth Division, P.O. Box 748, Mail Zone 2829, Fort Worth, TX 76101
51. R. Huse, Public Service Electric and Gas Company, 80 Park Place, Newark, NJ 07101
52. N. Kale, Roussel Engineering, Incorporated, 3017 Kingman Street, Metairie, LA 70002
53. H. A. Kamel, Department of Aerospace and Mechanical Engineering, College of Engineering, University of Arizona, Tucson, AR 85721
54. G. L. Kulcinski, Nuclear Engineering Department, 1500 Johnson Drive, University of Wisconsin, Madison, WI 53706
55. A. Kushner, Office of Naval Research, Department of the Navy, 800 Quincy Street, Code 474, Arlington, VA 22217
56. P. Landau, Single Buoy Moorings, Incorporated, Engineering Office, P.O. Box 157, 57 Rue Grimaldi, MC Monaco
57. E. Magrab, National Bureau of Standards, Receiving Room, Building 301, Route 270, Quince Orchard Road, Gaithersburg, MD 20760
58. A. M. Maloney, Electro-Motive Division, General Motors Corporation, Department 840, La Grange, IL 60525
59. S. Marsich, CETENA S.p.A., Viale IV Novembre 6, 16121 Genova, Italy
60. P. Mattenberger, c/o ITEP, EPFL-Department Genie Civil, 9 ch. des Delices, CH-1006 Lausanne, Switzerland
61. M. W. McCabe, Department of Aerospace and Mechanical Engineering, College of Engineering, University of Arizona, Tucson, AR 85721
62. R. McNaull, Maritime Administration, Department of Commerce Building, Washington, DC 20230
63. J. J. Nachtsheim, Maritime Administration, Department of Commerce Building, Washington, DC 20230
64. D. Nixon, Public Works Canada, 57 Sunnyside Avenue, Ottawa, Ontario, K1S 0P9, Canada
65. D. Radaj, Daimler-Benz, 700 Stuttgart 60, Postfach 202, Federal Republic of Germany
66. I. B. Richter, Computer Applications, Cummins Engine Company, Incorporated, Columbus, IN 47201
67. W. A. Rogers, General Dynamics, Fort Worth Division, P.O. Box 748, Mail Zone 2884, Fort Worth, TX 76101

68. H. J. Roussel, Roussel Engineering, Incorporated, 3017 Kingman Street, Metairie, LA 70002
69. R. Schmitz, Sperry Support Services, 1112 Church Street, Huntsville, AL 35801
70. D. R. Smith, Department of National Defence, 9 Grove Street, Box 1012, Dartmouth, Nova Scotia, B24 327, Canada
71. W. Spector, Department of Aerospace and Mechanical Engineering, College of Engineering, University of Arizona, Tucson, AR 85721
72. J. S. Spencer, U.S. Coast Guard, G-MMT-4/82, Washington, DC 20590
73. G. Stanley, Lockheed Aircraft, Palo Alto Research Laboratory, Department 52-33, Building 205, 3251 Hanover Street, Palo Alto, CA 94304
74. W. Wright, Public Works Canada, 57 Sunnyside Avenue, Ottawa, Ontario, K1S 0P9, Canada
75. Office of Assistant Manager for Energy Research and Development, Department of Energy, Oak Ridge Operations Office, Oak Ridge, TN 37830

76-258. Given distribution as shown in TID-4500, Mathematics and Computers (Distribution Category UC-32)

☆U.S. GOVERNMENT PRINTING OFFICE: 1980-640-245/423