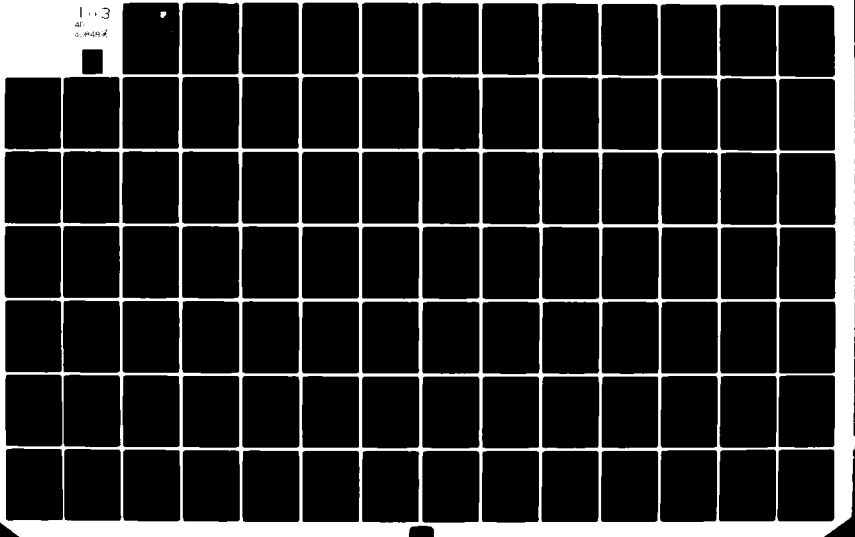
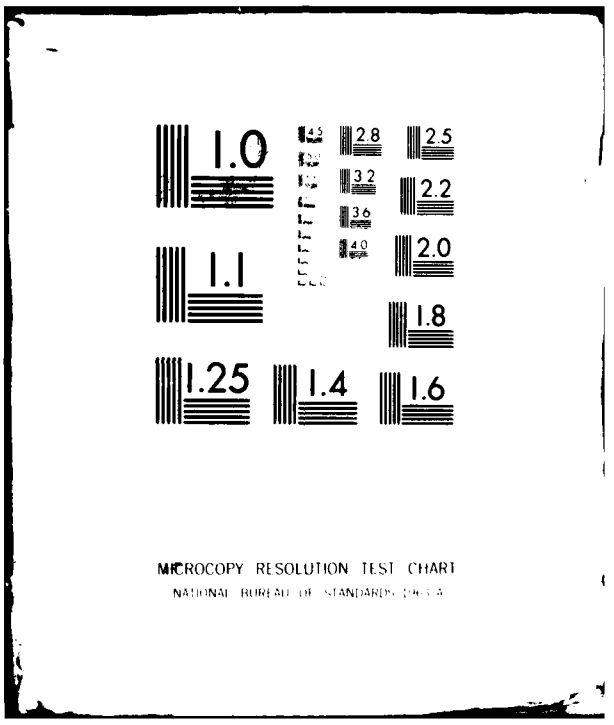
AD-A084 895 MEASUREMENT CONCEPT CORP ROME NY F/G 17/9
TACTICAL TARGET IDENTIFICATION (TTI) LABORATORY SIMULATION. VOL--ETC(U)
MAR 80 R KELLOGG, R A MCKIE, J TIMS F30602-79-C-0029
UNCLASSIFIED RADC -TR-80-41-VOL-1 NL




1.0
1.1
1.25
1.4
1.6
1.8
2.0
2.2
2.5
2.8
3.2
3.6
4.0
MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS 1963-A

AD A084895

[redacted]

TTI LABORATORY SIMULATION

Measurement Concept Corporation

Richard Kellogg
Robert A. McKie
James Tims

DTIC
ELECTE
MAY 27 1980
S D

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

THIS DOCUMENT IS BEST QUALITY PRACTICABLE.
THE COPY FURNISHED TO DDC CONTAINED A
SIGNIFICANT NUMBER OF PAGES WHICH DO NOT
REPRODUCE LEGIBLY.

ROME AIR DEVELOPMENT CENTER
Air Force Systems Command
Griffiss Air Force Base, New York 13441

DTIC FILE COPY

80 5 27 231

[redacted]

This report has been reviewed by the RADC Public Affairs Office and is releasable to the National Technical Information Service (NTIS). At NTIS it will be releasable to the general public, including foreign nations.

RADC-TR-80-41, Volume I (of two) has been reviewed and is approved for publication.

APPROVED: Richard J. Wood

RICHARD J. WOOD
Project Engineer

APPROVED: Frank J. Rehm

FRANK J. REHM
Technical Director
Surveillance Division

FOR THE COMMANDER: John P. Huss

JOHN P. HUSS
Acting Chief, Plans Office

If your address has changed or if you wish to be removed from the RADC mailing list, or if the addressee is no longer employed by your organization, please notify RADC (OCTM) Griffiss AFB NY 13441. This will assist us in maintaining a current mailing list.

Do not return this copy. Retain or destroy.

DISCLAIMER NOTICE

THIS DOCUMENT IS BEST QUALITY PRACTICABLE. THE COPY FURNISHED TO DTIC CONTAINED A SIGNIFICANT NUMBER OF PAGES WHICH DO NOT REPRODUCE LEGIBLY.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

(19) REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER (18) RADC-TR-80-41-Vol-1 (of two)	2. GOVT ACCESSION NO. AD-A081895
3. RECIPIENT'S CATALOG NUMBER	
4. TITLE (and Subtitle) (6) Tactical Target Identification (TTI) LABORATORY SIMULATION, Volume I.	5. TYPE OF REPORT & PERIOD COVERED (9) Final Technical Report, Nov 78 — Oct 79
6. PERFORMING ORG. REPORT NUMBER N/A	
7. AUTHOR(s) (10) Richard Kellogg, Robert A. McKie, James Tims	8. CONTRACT OR GRANT NUMBER(s) (15) F30602-79-C-0029 new
9. PERFORMING ORGANIZATION NAME AND ADDRESS Measurement Concept Corporation Rome NY 13440	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS (16) 62702F 45061341 (12) (13)
11. CONTROLLING OFFICE NAME AND ADDRESS Rome Air Development Center (OCTM) Griffiss AFB NY 13441	12. REPORT DATE (11) Mar 80
	13. NUMBER OF PAGES 269
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office) Same (12) 278	15. SECURITY CLASS. (of this report) UNCLASSIFIED
	15a. DECLASSIFICATION/DOWNGRADING SCHEDULE N/A

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

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

Same

18. SUPPLEMENTARY NOTES

RADC Project Engineer: Richard J. Wood (OCTM)

Vol II of this report will not be published for routine (see reverse)

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

Simulation	Image	Conversions
Processing	Interactive	
Identification	Programs	
Radar	Display	

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

Under the subject contract a general capability was developed to allow TTI Laboratory users to communicate with the COMTAL Image Processor, both under the DEC operating systems RT-11 and RSW-11M. This was specifically accomplished in three phases. First, the COMTAL Vision One Image Processing System was made accessible to the PDP-11/40 Processor under the RT-11 single-user operating system. Second, the COMTAL Image Processor was made to operate under the RSW-11M multi-user →

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

393 365

operating system, and was then integrated with the existing radar image processing system. Third, the radar return and air frame display program, Silhouette, was successfully converted to RSX-11M, and software written to enable the COMTAL image display to replace the previously supported DEC VT-11 Calligraphic Display.

Block 18 Cont'd

distribution but is available from RADC (OCTM), Griffiss AFB NY 13441 upon request. Volume II contains only the program listings as pertains to this volume.

PREFACE

This document describes technical services provided to Rome Air Development Center by the Measurement Concept Corporation under Contract F30602-79-C-0029. Three distinct efforts were performed by Mc^2 to integrate the COMTAL Vision One Image Display System with existing tools present in RADC's TTI (Tactical Target Identification) laboratory, located in Building 112 at GAFB. The Project Engineer of the subject effort was Mr. Richard Wood.

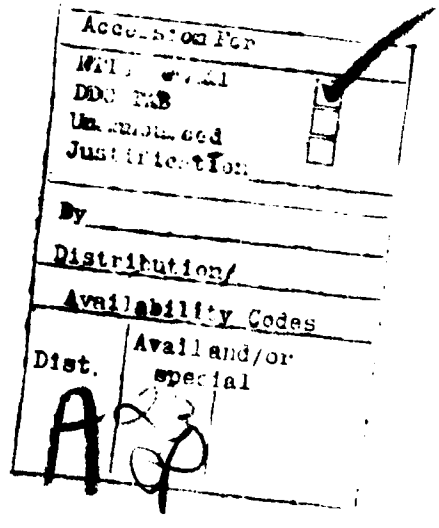


TABLE OF CONTENTS

		Page
1.0	RT-11COMTAL TO PDP-11 COMMUNICATIONS UTILITY.	1-1
1.1	COMPIP.FOR	1-2
1.2	CPIPCB.FOR	1-2
1.3	CPIPLB.FOR	1-2
1.4	COMLØ1.FOR	1-3
1.5	COMLØ2.FOR	1-3
1.6	COMLØ3.FOR	1-4
1.7	COMLØ4.FOR	1-4
1.8	COMLØ5.FOR	1-5
1.9	Instructions for Building the COMPIP Task	1-5
1.10	To Run COMPIP	1-6
2.0	INTEGRATION OF THE COMTAL VISION ONE WITH THE RIPS (RADAR) IMAGE PROCESSING SYSTEM	2-1
2.1	FRAMES.SCR and FRAMES.IPS	2-3
2.2	CALSUB.FTN	2-4
2.3	EXEC.ODL	2-4
2.4	PIPSB.FTN	2-4
2.5	COMLØ1.FOR	2-5
2.6	COMLØ2.FOR	2-5
2.7	COMLØ3.FOR	2-6
2.8	COMLØ4.FOR	2-6
2.9	COMLØ5.FOR	2-7
2.10	IWRITW.MAC	2-8
2.11	Building the Modified RIPS Task	2-9
2.12	To Run "RIPS" With The COMTAL	2-9
2.13	Integration of Mc^2 Software With Other User's Program	2-9

TABLE OF CONTENTS (Continued)

		Page
3.0	COMTAL: MC^2 VERSION OF THE SRC SILHOUETTE PROGRAM . .	3-1
3.1	Operation of the New SILHOUETTE Program	3-2
3.1.1	Preparation for Execution	3-2
3.1.2	Preparation for Execution of SILHOUETTE Program . . .	3-4
3.1.3	Operator Interaction with the SILHOUETTE Program. . .	3-5
3.1.4	Building "COMTAL", The Mc^2 Version of "SILHOUETTE". .	3-7
3.2	Modification and Additions to SRC Software	3-7
3.2.1	Modifications to SLMAIN.FTN	3-7
3.2.2	Modification to SLSUBS.FTN	3-10
3.2.3	COMGRF.FTN	3-11
3.2.4	Subroutine ROTATE	3-13
3.2.5	VECTOR.FTN	3-14
3.2.6	Other Programs	3-14

SUMMARY

The technical services provided by Mc^2 to RADC under the subject contract added three tools to the Tactical Target Identification Labratory (TTIL) in Building 112. First, the COMTAL Vision One Image Processing System was made accessible to the PDP-11/40 processor under the RT-11 Single-User operating system. Second, the COMTAL Vision One was made to operate under the RSX-11M Multi-User Operating System, and was then integrated with the existing PAR-developed Radar Image Processing System. Third, the Syracuse Research Corporation's Radar Return and Airframe Display Program, SILHOUETTE, was successfully converted to RSX-11M, and software was written to allow the COMTAL device to replace the previously supported DEC VT-11 Calligraphic Display.

In addition to these specific demonstrable tools, Mc^2 is leaving behind a general capability that will allow future TTIL users to easily communicate with the COMTAL system, both under RT-11 and under RSX-11M.

The following documentation describes the three main efforts under this contract. Copies of all software discussed reside both on RK-05 disk packs, and on backup tape volumes.

Many of the difficult problems encountered under this contract required trial-and-error solution, and the future avoidance of these problems by future TTIL users is an important legacy of this contract

Because of the disjoint nature of the three tasks funded by this contract, this Final Technical Report will be divided into three sections, each describing a complete task. Interrelationships between these tasks, where they exist, are duly noted, and there is a certain amount of back-and-forward referencing between sections. The three subtasks were carried

out chronologically in the order presented. Each task was executed by a different member of Mc^2's technical staff. The assignments are as follows:

COMPIP:	Richard Kellogg
RIPS, RSX and COMTAL Conversion:	James Tims
SILHOUETTE Conversion:	Robert A. McKie

In conclusion, Mc^2 would like to express its appreciation of the assistance given by the staff of the TTI Laboratory, Mr. P. Grasso, Mr. N. Internicola, and particularily Mr. Ronald Blackall, who was a concerned source of unwritten documentation, and a co-equal contributor to the three efforts.

Robert A. McKie, Project Manager
Measurement Concept Corporation

EVALUATION

The Tactical Target Identification (TTI) Laboratory/Simulation program was instituted to exploit the hardware/software resources of the TTI Laboratory. The evolution of effective peripherals for Target Identification Simulation and Real Data Processing, required an equivalent upgrade of software capabilities, which included modifications and development of an effective operating system controller. This effort provided for the combining of various independent programs for more effective operator control, and established existing capabilities from one hardware system to a more effective peripheral. The capabilities of the TTI Laboratory have thus been enhanced in performance, adaptability to new programs, and ease of understanding for future TTI applications. This effort was in accordance with TPO R2E to establish simulation capabilities for evaluation and validation of system programs for TTI assessments.

RICHARD J. WOOD
Project Engineer

1.0 RT-11 COMTAL TO PDP-11 COMMUNICATIONS UTILITY

The initial major task of the subject contract involved constructing a software package that would utilize the COMTAL DOS (Disk Operating System) device driver. This driver was modified by R. Kellogg to be usable under RT-11, but was tested initially on single lines of data. The objective of the initial effort was to construct a utility that would permit transfer of existing RIPS (Radar Image Processing System) files to the COMTAL for display.

The resulting COMPIP (COMTAL PIP) system served a twofold purpose. First, it allowed verification and demonstration of the VISION ONE by providing it a source of "real" data. Second, COMPIP was designed in modules that would be relatively easy to convert to RSX-11M. This design supported the objective of adapting the RIPS system to use the COMTAL for graphic output which, in fact, was the second major task undertaken, described in Section 2.0.

As constructed, COMPIP is a stand-alone program that functions much the same as the DEC PIP (Peripheral Interchange Program). It displays a list of COMTAL display options, requests a user selection from this list, then prompts the user into entering both file identification information and COMTAL memory identification information. Once entered, the information so described is written to or from the COMTAL.

Although not designed to be used as a tool integratable with a user application, COMPIP can indeed be used as part of a user program by removing the COMPIP.FOR main program, and building the remaining modules into the user's task.

The following paragraphs describe the functions of each of the software modules contained in COMPIP.

1.1 COMPIP.FOR

Author: R. Kellogg, Measurement Concept Corp., 1979.

COMPIP is the main program of the COMPIP task. It uses FORTRAN Read/Write statements to query the user regarding which COMTAL function he would like performed, then directs passage of control to the proper subroutine contained in CPIPSB.FOR. This program is essentially a "dummy" main calling program.

1.2 CPIPSB.FOR

CPIPSB.FOR contains the main subroutines for the program COMPIP. Each subroutine is a self-contained main section, and calls several of the COMTAL library routines, as well as other IPS file reading routines. The main routines in CPIPSB.FOR are:

IWIMF - Read an IPS image file to the COMTAL.
IRIMF - Write an IPS image file from the COMTAL
IWGMF - Read an IPS graphics file to the COMTAL.
IRGMF - Write an IPS graphics file from the COMTAL.
IWFMF - Read an IPS function file to the COMTAL.
IRFMF - Write an IPS function file from the COMTAL.
IWPSF - Read an IPS pseudo-color file to the COMTAL.
IRPSF - Write an IPS pseudo-color file from the COMTAL.

1.3 CPIPLB.FOR

CPIPLB.FOR is a collection of subroutines that open, close, read and write various IPS files for use with COMPIP. When COMPIP is later re-written to be integrated as a RIPS "frame" option, the module will not be necessary.

The routines contained in CPIPLB.FOR are:

IPROPE - Opens an IPS file for reading.
IPWOPE - Opens an IPS file for output.
IPRDIM - Reads one record from an IPS file to the user's buffer.
IPWTIM - Writes one record from the user's buffer to an IPS file.
IPCLOS - Closes an IPS file.
IPRDPS - Reads one Pseudo-color Memory record into user buffer.
IPWTPS - Writes a Pseudo-color Memory record to an IPS file.
IPRDGM - Reads one line of Graphics Memory from an IPS file into the user buffer.
IPWTGM - Writes one Graphic Memory record to an IPS file.
IPRDFM - Reads one Function Memory record from an IPS file to the user buffer.
IPWTFM - Writes one Function Memory record to an IPS file.

1.4 COMLØ1.FOR

COMLØ1.FOR is the first of five COMTAL library modules containing routines that support reading and writing to and from the COMTAL. COMLØ1 establishes some data definitions, common areas, data equivalences and COMTAL command word tables. It also contains two routines to open and close the COMTAL's I/O channel:

ICMOPE - Opens channel for COMTAL service.
ICMCLO - Closes the COMTAL channel.

1.5 COMLØ2.FOR

COMLØ2.FOR contains routines to read and write COMTAL "Image" and "Graphic" memories.

IWIM - Writes one line of data from user memory to selected COMTAL Image Memory.

IRIM - Reads one line of data from selected COMTAL Image Memory to user buffer.

IWGM - Writes one line of data from user memory to the selected COMTAL Graphic Memory.

IRGM - Reads one line of data from the selected COMTAL Graphic Memory to the user buffer.

ITC1 - Performs the actual "Transfer Code 1" command for the above four routines.

1.6 COML03.FOR

COML03.FOR contains routines to read and write COMTAL Function and Pseudo-color Memory. It contains the following routines:

IWFM - Writes to the selected COMTAL Function Memory from the user buffer.

IRFM - Reads the selected Function Memory into the user buffer.

IWPS - Writes to the selected Pseudo-color Memory from the user buffer.

IRPS - Reads from the selected Pseudo-color Memory into the user buffer.

ITC2 - Performs the actual "Transfer Code 2" command for the above four routines.

1.7 COML04.FOR

COML04.FOR contains routines to read and write the location of the COMTAL cursor. There are only two subroutines in this module. They are:

WTP - Writes X, or Y, or both to the target cursor in the COMTAL.

RTP - Reads the target cursor X, Y location into variables IX, IY.

1.8 COMLØ5.FOR

COMLØ5.FOR contains routines to read and write blocks of text to and from the COMTAL, read and write the data block to and from the COMTAL, and read and write blocks of contiguous memory between the COMTAL and the host computer. The routines are:

- WCB - Transmit a block of text to the COMTAL text buffer.
- RCB - Read a block of text from the COMTAL text buffer.
- WDB - Transmit a Data Block to the COMTAL. The Data Block contains the contents of several display registers that determine the status of the COMTAL display.
- RDB - Reads the COMTAL Data Block.
- WCD - This routine is used to write any block of memory in the COMTAL. It primary use is down line loading code to the COMTAL, when COMTAL memory is acquired to support this loading.
- RCD - This routine reads any block of memory from the COMTAL.
- TC3 - This routine issues the "Transfer Code 3" command for the above six routines.

1.9 Instructions for Building the COMPIP Task

The following RT-11 commands are necessary to instruct the RT-11 Linker to link together the various modules comprising COMPIP into an executable task image:

```
.R LINK
*COMPIP = COMPIP, CPIPSB, CPIPLB, SYSLIB/F/C
*COMLØ1, COMLØ2, COMLØ3 (, COMLØ4, COMLØ5)
```

1.10 To Run COMPIP

.RUN COMPIP

COMPIP will prompt the user for necessary input.

```
C     .                 FILE NAME: COMPIP.FOR
C     .
C     .                 DATE LAST EDITED: 18-APR-79
C     .
C     .                 HISTORY:
C     .
C     .                 DATE:           MODIFICATION
C     .                 -----           ------------
C     .                 19-MAR-79       INITAL WRITING
C     .                 21-MAR-79       ADD LINK INSTRUCTIONS
C     .                 18-APR-79       ADD OPTION 'IWCBF'
C     .
C     .    THIS PROGRAM WILL DISPLAY A MENU TO READ/WRITE IPS
C     .    FORMAT FILES AND WRITE/READ THE COMTAL.
C     .
C     .    THIS ROUTINE WILL CALL THE APPROPRIATE ROUTINE TO DO
C     .    THE NECESSARY FUNCTION.  IT IS SET UP SIMILAR TO THE
C     .    IPS ROUTINE ' CALSUB ' USED FOR THE SAME FUNCTION IN IPS.
C     .    THIS WILL EASE TRANSFERING TO RSX AND IPS.
C     .
C     .    THIS PROGRAM IS FOR RT-11 ONLY AND A DIFFERENT ONE WILL
C     .    BE INCORPORATED INTO IPS FOR RSX.
C     .
C     .    TO COMPILE:
C     .    -- -------
C     .
C     .    .R FORTRA
C     .    *COMPIP =COMPIP/U
C     .
C     .    TO LINK:
C     .    -- ----
C     .
C     .    *COMPIP = COMPIP,CPIPSB,CPIPLB,SYSLIB/F/C
C     .    *COML01,COML02,COML03 ( ,COML04,COML05 )
C     .
C     .    1) QUERY USER FOR READ OR WRITE
C     .    2) CALL APROPRIATE ROUTINE
C     .    3) REPORT ERRORS
C     .    4) DO AGAIN
C
C     .    1) QUERY USER
```

```
C
C
0001   50    WRITE(7,2000)
0002   2000  FORMAT(' INPUT DESIRED ACTION '//,
     *     1) READ   AN IPS IMAGE     FILE AND   DISPLAY  ON COMTAL',/,
     *       2) WRITE  AN IPS IMAGE     FILE FROM          THE COMTAL',/,/,
     *     3) READ   AN IPS GRAPHICS  FILE AND   DISPLAY  ON COMTAL',/,
     *       4) WRITE  AN IPS GRAPHICS  FILE FROM          THE COMTAL',/,/,
     *     5) READ   AN IPS FUNCTION  FILE AND   WRITE    TO  COMTAL',/,
     *       6) WRITE  AN IPS FUNCTION  FILE FROM          THE COMTAL',/,/,
```



```
     *     7) READ   AN IPS PSEUDO    FILE AND   WRITE    TO  COMTAL',/,
     *       8) WRITE  AN IPS PSEUDO    FILE FROM          THE COMTAL',/,/,
     *     9) WRITE  COMMAND BLOCK                       TO  COMTAL',/,/,
     *                DESIRED ACTION: ',$ )
      C
0003        READ(7,110)IRW
0004  110   FORMAT(2I6)
      C
      C
      C
      C
      C     ,    2) DISPATCH TO ROUTINE
      C
      C
      C
0005        IF( IRW .EQ.  1 )  CALL IWIMF
0007        IF( IRW .EQ.  2 )  CALL IRIMF
0009        IF( IRW .EQ.  3 )  CALL IWGMF
0011        IF( IRW .EQ.  4 )  CALL IRGMF
0013        IF( IRW .EQ.  5 )  CALL IWFMF
0015        IF( IRW .EQ.  6 )  CALL IRFMF
0017        IF( IRW .EQ.  7 )  CALL IWPSF
0019        IF( IRW .EQ.  8 )  CALL IRPSF
0021        IF( IRW .EQ.  9 )  CALL IWCBF
      C
      C
0023        IF(( IRW .GT.9) .OR.(IRW.LT.1)) GO TO 1000
      C
      C     ,    4) DO AGAIN
      C
0025        GO TO 50
      C
      C
      C
      C     ,    3) ERROR
      C
0026  1000  WRITE(7,1100)
0027  1100  FORMAT(' FUNCTION NOT IMPLIMENTED ' )
```

```
0025          GO TO 50
0026          END
```

```
RT-11 FORTRAN IV        STORAGE MAP

NAME    OFFSET  ATTRIBUTES

IRU     001204  INTEGER*2 VARIABLE
IUIMF   000000  INTEGER*2 PROCEDURE
IRIMF   000000  INTEGER*2 PROCEDURE
IUOMF   000000  INTEGER*2 PROCEDURE
IROMF   000000  INTEGER*2 PROCEDURE
IWFMF   000000  INTEGER*2 PROCEDURE
IRFMF   000000  INTEGER*2 PROCEDURE
IUPSF   000000  INTEGER*2 PROCEDURE
IRPSF   000000  INTEGER*2 PROCEDURE
IUCBF   000000  INTEGER*2 PROCEDURE
*
```

```
C     .                    FILE NAME: CPIPSB.FOR
C     .
C     .                    DATE LAST EDITED: 18-APR-79
C     .
C     .                    HISTORY:
C     .
C     .                    DATE:          MODIFICATION
C     .                    -----          ------------
C     .                    19-MAR-79      INITAL WRITING
C     .                    21-MAR-79      ADD MORE SUBS
C     .                    18-APR-79      ADD CLEAR SCREEN
C     .                                   TO 'IWIMF', AND
C     .                                   ADD 'IWCBF'
C     .
C     .    THIS FILE CONTAINS THE MAIN SUBROUTINES FOR THE
C     .    PROGRAM 'COMPIP'. EACH SUBROUTINE IS A SELF CONTAINED
C     .    MAIN SECTION AND CALLS SEVERAL OF THE COMTAL LIBRARY
C     .    ROUTINES AS WELL AS OTHER IPS FILE READING ROUTINES.
C     .
C     .    THE MAIN ROUTINES ARE:
C     .
C     .            IWIMF  -  READ AN IPS FILE DISPLAY ON COMTAL
C     .            IRIMF  -  WRITE AN IPS FILE FROM THE COMTAL
C     .            IWGMF  -  READ AN IPS GRAPHICS FILE TO COMTAL
C     .            IRGMF  -  WRITE AN IPS GRAPHICS FILE FROM COMTAL
C     .            IWFMF  -  READ AN IPS FUNCTION FILE TO COMTAL
C     .            IRFMF  -  WRITE AN IPS FUNCTION FILE FROM COMTAL
C     .            IWPSF  -  READ AN IPS PSEUDO-COLOR FILE TO COMTAL
C     .            IRPSF  -  WRITE AN IPS PSEUDO-COLOR FILE FROM COMTAL
C     .            IWCBF  -  WRITE COMMAND BLOCK TO COMTAL
C
0001        SUBROUTINE IWIMF
C     .
C     .    FUNCTIONAL DESCRIPTION:
C     .
C     .    THIS ROUTINE WILL READ AN IPS 'IMAGE' TYPE FILE AND
C     .    WRITE IT TO ONE OF THE IMAGE MEMORIES IN THE COMTAL.
C     .
C     .    1) QUERY USER FOR IMAGE NUMBER
C     .    2) OPEN FILE FOR READ/WRITE
C     .    3) OPEN COMTAL FOR PROCESSING
```

```
C     ,    4) CLEAR SCREEN
C     ,    5) READ FILE, WRITE COMTAL
C
0002        LOGICAL*1 ITEXT(8),ONE,TWO,THREE,FOUR
C
0003        INTEGER NFTYP(2)
0004        INTEGER IBUFF(512)
0005        COMMON /LINE/ IBUFF
C
0006        DATA ITEXT /'C','L',' ','I',' ','1',' ','8'/
```

```
RT-11 FORTRAN IV        V01B-08A   WED 18-APR-79 18:12:34       PAGE 002

0007        DATA ONE /'1'/, TWO /'2'/, THREE /'3'/, FOUR /'4'/
C
0008        DATA NFTYP /3RIMG,3RIMG/
C
C
C     ,    1) QUERY USER
C
C
0009  50    WRITE(7,2000)
0010  2000  FORMAT(/,'  INPUT IMAGE NUMBER: ',$)
C
0011        READ(7,110)IMNO
0012  110   FORMAT(2I6)
C
C
0013        IF(IMNO.EQ.1) ITEXT(6) = ONE
0015        IF(IMNO.EQ.2) ITEXT(6) = TWO
0017        IF(IMNO.EQ.3) ITEXT(6) = THREE
0019        IF(IMNO.EQ.4) ITEXT(6) = FOUR
C
C
C
C     ,    READ IPS FILE, WRITE TO COMTAL
C
C     ,    2) OPEN IPS FILE
C     ,    3) OPEN COMTAL
C
C
C
C
0021        IER =  IPROPE(NRECS,NRECSZ,NFTYP)
C
C
C
0022        IEP=ICMOPE()
C
C     ,    CALC OFFSET TO PUT IN CENTER OF SCREEN
C
```

```
0023          IXOFF = ( 128 - NRECSZ ) /2
0024          IF(IXOFF.LT.1) IXOFF = 1
      C
0026          IYOFF = ( 256 - NRECS ) /2
0027          IF ( IYOFF .LT .1) IYOFF = 1
      C
      CCC     .    CLEAR OUT BUFFER
      C
0029          DO 900 J=1,256
      C
0030          IBUFF(J)= 0
      C
0031    900   CONTINUE
      CCC
      CCC     .    4) CLEAR IMAGE
      C
0032          IER = IWCB(ITEXT)
      C
```



```
      C
      CCC
      CCC     .    5) READ A IPS RECORD , WRITE TO COMTAL
      CCC
      D       WRITE (7,987)NRECS
      D 987   FORMAT(' NRECS = ',I8,' MAIN 2  JUST PRIOR DO LOOP READ/WRITE ')
      CCC
0033          DO 100 K=1,NRECS
      C
0034          J = K -1 + IYOFF
      CD
      CCC     PAUSE' MAIN DO JUST PRIOR TO IPREAD '
      C
0035          IER = IPRDIM( IBUFF(IXOFF) )
      C
      D       PAUSE ' MAIN DO JUST PRIOR TO IWIM '
      C
0036          IER=IWIM(IMNO,J,IBUFF)
      C
0037    100   CONTINUE
      C
0038    102   CONTINUE
0039    103   CONTINUE
      C
0040          CALL IPCLOS
0041          CALL ICMCLO
      C
```

```
0042        RETURN
0043        END
```

```
RT-11 FORTRAN IV        STORAGE MAP

NAME    OFFSET  ATTRIBUTES

ITEXT   000014  LOGICAL*1 ARRAY (8)
NFTYP   000024  INTEGER*2 ARRAY (2)
ONE     000030  LOGICAL*1 VARIABLE
TWO     000031  LOGICAL*1 VARIABLE
THREE   000032  LOGICAL*1 VARIABLE
FOUR    000033  LOGICAL*1 VARIABLE
IMNO    000076  INTEGER*2 VARIABLE
IER     000100  INTEGER*2 VARIABLE
IPROPE  000000  INTEGER*2 PROCEDURE
NRECS   000102  INTEGER*2 VARIABLE
NRECSZ  000104  INTEGER*2 VARIABLE
ICMOPE  000000  INTEGER*2 PROCEDURE
IXOFF   000106  INTEGER*2 VARIABLE
IYOFF   000110  INTEGER*2 VARIABLE
J       000112  INTEGER*2 VARIABLE
IWCS    000000  INTEGER*2 PROCEDURE
K       000114  INTEGER*2 VARIABLE
IPRDIM  000000  INTEGER*2 PROCEDURE
IWIM    000000  INTEGER*2 PROCEDURE
IPCLOS  000000  INTEGER*2 PROCEDURE
ICMCLO  000000  INTEGER*2 PROCEDURE

COMMON BLOCK /LINE/     LENGTH 002000

IBUFF   000000  INTEGER*2 ARRAY (512)
```



```
RT-11 FORTRAN IV        V01B-08A   WED 18-APR-79 18:12:38       PAGE 001

                C
0001            SUBROUTINE IRIMF
                C
                C     ,    FUNCTIONAL DESCRIPTION:
                C
                C     ,    THIS ROUTINE WILL READ A COMTAL IMAGE AND
                C     ,    WRITE IT TO AN IPS ' IMAGE '  FILE.
                C
                C
```

```
C     ,    1) QUERY USER FOR IMAGE NUMBER
C     ,    2) OPEN FILE FOR READ/WRITE
C     ,    3) OPEN COMTAL FOR PROCESSING
C     ,    4) READ COMTAL, WRITE FILE
C
      INTEGER NFTYP(2),NAFTYP(2)
      INTEGER IBUFF(512)
      COMMON /LINE/ IBUFF
C
      DATA NFTYP /3RIMG,3RIMG/, NAFTYP /'.I','MG'/
C
CCC
C     ,    1) QUERY USER
CCC
C
50    WRITE(7,2000)
2000  FORMAT(/,'  INPUT IMAGE NUMBER: ',$)
C
      READ(7,110)IMNO
110   FORMAT(2I6)
C
CCCCC
C     ,    READ COMTAL ,WRITE IPS FILE
CCCCC
C
1020  CONTINUE
C
CD    PAUSE ' MAIN 3  JUST PRIOR IPWOPE '
CCC
C
      IER =  IPWOPE(256,256,NFTYP,NAFTYP)
C
      IER=ICMOPE()
C
CD    PAUSE ' MAIN 4 JUST PRIOR DO LOOP WRITE/READ '
CC
      DO 200      J=1,256
C
      IER=IRIM(IMNO,J,IBUFF)
C
      IER =  IPUTIM(IBUFF)
C
200   CONTINUE
C
      CALL IPCLOS
      CALL ICMCLO
```

```
      C
0019          RETURN
0020          END
```

```
RT-11 FORTRAN IV        STORAGE MAP

NAME    OFFSET  ATTRIBUTES

NFTYP   000014  INTEGER*2 ARRAY (2)
NAFTYP  000020  INTEGER*2 ARRAY (2)
IMNO    000070  INTEGER*2 VARIABLE
IER     000072  INTEGER*2 VARIABLE
IPUOPE  000000  INTEGER*2 PROCEDURE
ICMOPE  000000  INTEGER*2 PROCEDURE
J       000074  INTEGER*2 VARIABLE
IRIM    000000  INTEGER*2 PROCEDURE
IPUTIM  000000  INTEGER*2 PROCEDURE
IPCLOS  000000  INTEGER*2 PROCEDURE
ICMCLO  000000  INTEGER*2 PROCEDURE

COMMON BLOCK /LINE/     LENGTH 002000

IBUFF   000000  INTEGER*2 ARRAY (512)
```



```
RT-11 FORTRAN IV        V01B-08A   WED 18-APR-79 18:12:41       PAGE 001

      C
0001          SUBROUTINE IWPSF
      C
      C     .    FUNCTIONAL DESCRIPTION:
      C
      C     .    THIS ROUTINE WILL READ AN IPS PSEUDO-COLOR FILE
      C     .    AND WRITE IT TO THE COMTAL PSEUDO-COLOR MEMORY.
      C
      C     .    1)
      C     .    2) OPEN FILE FOR READ/WRITE
      C     .    3) OPEN COMTAL FOR PROCESSING
      C     .    4) READ FILE, WRITE COMTAL
      C     .    5) CLOSE COMTAL, CLOSE IPS FILE
      C
0002          INTEGER NFTYP(2)
0003          INTEGER IBUFF(512)
```

```
0004          COMMON /LINE/ IBUFF
0005      C   DATA NFTYP /3RPSE,3RPSE/
          C
          C
          C   ,    1)
          C
          C
          C   ,    2)
          C
0006          IER = IPROPE(NRECS,NRECSZ,NFTYP)
          C
          C   ,    3) OPEN COMTAL
          C
0007          IER=ICMOPE()
          C
          C
          C   ,    4) READ IPS FILE, WRITE TO COMTAL PSEUDO-COLOR
          C
0008          IER = IPRDPS (IBUFF)
          C
0009          IER = IWPS (IBUFF)
          C
          C   ,    5) CLOSE IPS FILE, CLOSE COMTAL
          C
0010          CALL IPCLOS
          C
0011          IER = ICMCLO
          C
          C
0012          RETURN
0013          END
```

```
RT-11 FORTRAN IV        STORAGE MAP

NAME    OFFSET  ATTRIBUTES

NFTYP   000014  INTEGER*2 ARRAY (2)
IER     000020  INTEGER*2 VARIABLE
IPROPE  000000  INTEGER*2 PROCEDURE
NRECS   000022  INTEGER*2 VARIABLE
NRECSZ  000024  INTEGER*2 VARIABLE
ICMOPE  000000  INTEGER*2 PROCEDURE
IPRDPS  000000  INTEGER*2 PROCEDURE
IWPS    000000  INTEGER*2 PROCEDURE
IPCLOS  000000  INTEGER*2 PROCEDURE
ICMCLO  000026  INTEGER*2 VARIABLE

COMMON BLOCK /LINE/     LENGTH 002000

IBUFF   000000  INTEGER*2 ARRAY (512)
```

```
      C
0001      SUBROUTINE IRPSF
      C
      C     ,    FUNCTIONAL DESCRIPTION:
      C
      C     ,    THIS ROUTINE WILL READ THE COMTAL PSEUDO-COLOR MEMORY
      C     ,    AND WRITE IT TO AN IPS PSEUDO-COLOR FILE.
      C
      C
      C     ,    1)
      C     ,    2) OPEN FILE FOR READ/WRITE
      C     ,    3) OPEN COMTAL FOR PROCESSING
      C     ,    4) READ COMTAL, WRITE FILE
      C     ,    5) CLOSE COMTAL, CLOSE IPS FILE
      C
      C
0002      INTEGER NFTYP(2),NAFTYP(2)
0003      INTEGER IBUFF(512)
0004      COMMON /LINE/ IBUFF
      C
0005      DATA NFTYP /3RPSE,3RPSE/, NAFTYP /'.P','SE'/
      C
      C
      C     ,    1)
      C
      C
      C     ,    2)
      C
0006      NRECS = 1
0007      NRECSZ= 256
      C
0008      IER = IPUOPE(NRECS,NRECSZ,NFTYP,NAFTYP)
      C
      C     ,    3) OPEN COMTAL
      C
0009      IER=ICMOPE()
      C
      C
      C     ,    4) READ COMTAL, WRITE TO IPS PSEUDO-COLOR
      C
0010      IER = IRPS (IBUFF)
      C
0011      IER = IPUTPS (IBUFF)
      C
      C
      C     ,    5) CLOSE IPS FILE, CLOSE COMTAL
      C
0012      CALL IPCLOS
```

```
C
0013        IER = ICMCLO
C
C
0014        RETURN
0015        END
```

```
RT-11 FORTRAN IV        STORAGE MAP

NAME    OFFSET  ATTRIBUTES

NFTYP   000014  INTEGER*2 ARRAY (2)
NWFTYP  000020  INTEGER*2 ARRAY (2)
NRECS   000024  INTEGER*2 VARIABLE
NRECSZ  000026  INTEGER*2 VARIABLE
IEP     000030  INTEGER*2 VARIABLE
IPWOPE  000000  INTEGER*2 PROCEDURE
ICMOPE  000000  INTEGER*2 PROCEDURE
IRPS    000000  INTEGER*2 PROCEDURE
IPWTPS  000000  INTEGER*2 PROCEDURE
IPCLOS  000000  INTEGER*2 PROCEDURE
ICMCLO  000032  INTEGER*2 VARIABLE

COMMON BLOCK /LINE/     LENGTH 002000

IBUFF   000000  INTEGER*2 ARRAY (512)
```



```
RT-11 FORTRAN IV        V01B-08A   WED 18-APR-79 18:12:46        PAGE 001

0001        SUBROUTINE IWGMF
C
C       FUNCTIONAL DESCRIPTION:
C
C       THIS ROUTINE WILL READ AN IPS GRAPHICS (.GRA) FILE
C       AND WRITE IT TO THE COMTAL GRAPHICS MEMORY.
C
C
C       1)
C       2) OPEN FILE FOR READ/WRITE
C       3) OPEN COMTAL FOR PROCESSING
C       4) READ FILE, WRITE COMTAL
C       5) CLOSE COMTAL, CLOSE IPS FILE
C
```

```
                C
0002                INTEGER NFTYP(2)
0003                INTEGER IBUFF(512)
0004                COMMON /LINE/ IBUFF
                C
0005                DATA NFTYP /3RGRA,3RGRA/
                C
                C
                C     ,    1) QUERY USER FOR GRAGHICS NUMBER
                C
0006                WRITE(7,110)
0007    110         FORMAT(' INPUT GRAGHICS NUMBER: ',$)
0008                READ(7,120) IMNO
0009    120         FORMAT(I6)
                C
                C
                C     ,    2)
                C
0010                IER = IPROPE(NRECS,NRECSZ,NFTYP)
                C
                C     ,    3) OPEN COMTAL
                C
0011                IER=ICMOPE()
                C
                C
                C     ,    4) READ IPS FILE, WRITE TO COMTAL GRAPHICS MEMORY
                C
0012                DO 100 J=1,256
                C
0013                IER = IPRDGM (IBUFF)
                C
0014                IER = IWGM (IMNO,J,IBUFF)
                C
0015    100         CONTINUE
                C
                C     ,    5) CLOSE IPS FILE, CLOSE COMTAL
                C
0016                CALL IPCLOS
                C
0017                IER = ICMCLO
                C
                C
```



```
0018                RETURN
0019                END
```

```
RT-11 FORTRAN IV        STORAGE MAP

NAME    OFFSET  ATTRIBUTES

NFTYP   000014  INTEGER*2 ARRAY (2)
IMNO    000062  INTEGER*2 VARIABLE
IER     000064  INTEGER*2 VARIABLE
IPROPE  000000  INTEGER*2 PROCEDURE
NRECS   000066  INTEGER*2 VARIABLE
NRECSZ  000070  INTEGER*2 VARIABLE
ICMOPE  000000  INTEGER*2 PROCEDURE
J       000072  INTEGER*2 VARIABLE
IPRDGM  000000  INTEGER*2 PROCEDURE
IWGM    000000  INTEGER*2 PROCEDURE
IPCLOS  000000  INTEGER*2 PROCEDURE
ICMCLO  000074  INTEGER*2 VARIABLE

COMMON BLOCK /LINE/     LENGTH 002000

IBUFF   000000  INTEGER*2 ARRAY (512)
```



```
RT-11 FORTRAN IV        V01B-08A   WED 18-APR-79 18:12:49       PAGE 001

                C
0001                 SUBROUTINE IRGMF
                C
                C    ,    FUNCTIONAL DESCRIPTION:
                C
                C    ,    THIS ROUTINE WILL READ THE COMTAL GRAPGICS MEMORY
                C    ,    AND WRITE IT TO AN IPS GRAPHICS (.GRA) FILE.
                C
                C
                C    ,    1) QUERY USER FOR GRAPHICS NUMBER
                C    ,    2) OPEN FILE FOR READ/WRITE
                C    ,    3) OPEN COMTAL FOR PROCESSING
                C    ,    4) READ COMTAL, WRITE FILE
                C    ,    5) CLOSE COMTAL, CLOSE IPS FILE
                C
                C
0002                 INTEGER NFTYP(2),NAFTYP(2)
0003                 INTEGER IBUFF(512)
0004                 COMMON /LINE/ IBUFF
                C
0005                 DATA NFTYP /3RGRA,3RGRA/, NAFTYP /'.G','RA'/
                C
                C
                C    ,    1) QUERY USER FOR GRAGHICS NUMBER
```

```
C
      WRITE(7,110)
  110 FORMAT(' INPUT GRAGHICS NUMBER: ',$)
      READ(7,120) IMNO
  120 FORMAT(I6)
C
C
C     ,    2)
C
      NRECS = 256
      NRECSZ= 16
C
      IER = IPWOPE(NRECS,NRECSZ,NFTYP,NAFTYP)
C
C     ,    3) OPEN COMTAL
C
      IER=ICMOPE()
C
C
C     ,    4) READ COMTAL, WRITE TO IPS GRAPHICS
C
      DO 100 J=1,256
C
      IER = IRGM (IMNO,J,IBUFF)
C
      IER = IPUTGM (IBUFF)
C
  100 CONTINUE
C
C
C     ,    5) CLOSE IPS FILE, CLOSE COMTAL
C
```



```
      CALL IPCLOS
C
      IER = ICMCLO
C
C
      RETURN
      END
```

```
RT-11 FORTRAN IV        STORAGE MAP

NAME    OFFSET  ATTRIBUTES

NFTYP   000014  INTEGER*2 ARRAY (2)
NWFTYP  000020  INTEGER*2 ARRAY (2)
IMNO    000066  INTEGER*2 VARIABLE
NRECS   000070  INTEGER*2 VARIABLE
NRECS2  000072  INTEGER*2 VARIABLE
IER     000074  INTEGER*2 VARIABLE
IPWOPE  000000  INTEGER*2 PROCEDURE
ICMOPE  000000  INTEGER*2 PROCEDURE
J       000076  INTEGER*2 VARIABLE
IRGM    000000  INTEGER*2 PROCEDURE
IPUTGM  000000  INTEGER*2 PROCEDURE
IPCLOS  000000  INTEGER*2 PROCEDURE
ICMCLO  000100  INTEGER*2 VARIABLE

COMMON BLOCK /LINE/     LENGTH 002000

IBUFF   000000  INTEGER*2 ARRAY (512)
```



```
RT-11 FORTRAN IV        V01B-08A   WED 18-APR-79 18:12:52       PAGE 001
      C
0001        SUBROUTINE IWFMF
      C
      C     ,    FUNCTIONAL DESCRIPTION:
      C     ,
      C     ,    THIS ROUTINE WILL READ AN IPS FUNCTION (.TRA ) FILE
      C     ,    AND WRITE IT TO THE COMTAL FUNCTION  MEMORY.
      C     ,
      C     ,    1) QUERY USER FOR FUNCTION NUMBER
      C     ,    2) OPEN FILE FOR READ/WRITE
      C     ,    3) OPEN COMTAL FOR PROCESSING
      C     ,    4) READ FILE, WRITE COMTAL
      C     ,    5) CLOSE COMTAL, CLOSE IPS FILE
      C
0002        INTEGER NFTYP(2)
0003        INTEGER IBUFF(512)
0004        COMMON /LINE/ IBUFF
      C
0005        DATA NFTYP /3RTRA,3RTRA/
      C
      C     ,    1) QUERY USER FOR FUNCTION NUMBER
      C
0006        WRITE(7,110)
```

```
0007   110   FORMAT(' INPUT FUNCTION NUMBER: ',$)
0008         READ(7,120) IMNO
0009   120   FORMAT(I6)
      C
      CCCC
      CCCC     ,    2)
      C
0010         IER = IPROPE(NRECS,NRECSZ,NFTYP)
      CCC      ,    3) OPEN COMTAL
      C
0011         IER=ICMOPE()
      C
      CCCC
      CCCC     ,    4) READ IPS FILE, WRITE TO COMTAL FUNCTION MEMORY
      C
0012         IER = IPRDFM (IBUFF)
      C
0013         IER = IWFM (IMNO,IBUFF)
      C
      CCC      ,    5) CLOSE IPS FILE, CLOSE COMTAL
      C
0014         CALL IPCLOS
      C
0015         IER = ICMCLO
      C
      CC
0016         RETURN
0017         END
```

```
RT-11 FORTRAN IV        STORAGE MAP

NAME     OFFSET  ATTRIBUTES

NFTYP    000014  INTEGER*2 ARRAY (2)
IMNO     000062  INTEGER*2 VARIABLE
IER      000064  INTEGER*2 VARIABLE
IPROPE   000000  INTEGER*2 PROCEDURE
NRECS    000066  INTEGER*2 VARIABLE
NRECSZ   000070  INTEGER*2 VARIABLE
ICMOPE   000000  INTEGER*2 PROCEDURE
IPRDFM   000000  INTEGER*2 PROCEDURE
IWFM     000000  INTEGER*2 PROCEDURE
IPCLOS   000000  INTEGER*2 PROCEDURE
ICMCLO   000072  INTEGER*2 VARIABLE

COMMON BLOCK /LINE/     LENGTH 002000

IBUFF    000000  INTEGER*2 ARRAY (512)
```

```
      C
0001        SUBROUTINE IRFMF
      C
      C     ,    FUNCTIONAL DESCRIPTION:
      C
      C     ,    THIS ROUTINE WILL READ THE COMTAL FUNCTION MEMORY
      C     ,    AND WRITE IT TO AN IPS TRANSFER FUNCTION (.TRA) FILE.
      C
      C     ,    1) QUERY USER FOR FUNCTION NUMBER
      C     ,    2) OPEN FILE FOR READ/WRITE
      C     ,    3) OPEN COMTAL FOR PROCESSING
      C     ,    4) READ COMTAL, WRITE FILE
      C     ,    5) CLOSE COMTAL, CLOSE IPS FILE
      C
0002        INTEGER NFTYP(2),NAFTYP(2)
0003        INTEGER IBUFF(512)
0004        COMMON /LINE/ IBUFF
      C
0005        DATA NFTYP /3RTRA,3RTRA/, NAFTYP /'.T','RA'/
      C
      C     ,    1) QUERY USER FOR FUNCTION NUMBER
      C
0006        WRITE(7,110)
0007  110   FORMAT(' INPUT FUNCTION NUMBER: ',$)
0008        READ(7,120) IFNO
0009  120   FORMAT(I6)
      C
      C     ,    2)
      C
0010        NRECS = 1
0011        NRECSZ= 128
      C
0012        IER = IPUOPE(NRECS,NRECSZ,NFTYP,NAFTYP)
      C
      C     ,    3) OPEN COMTAL
      C
0013        IER=ICMOPE()
      C
      C     ,    4) READ COMTAL, WRITE TO IPS TRANSFER FUNCTION FILE
      C
0014        IER = IRFM (IFNO,IBUFF)
      C
0015        IER = IPUTFM (IBUFF)
```

```
C     ,    5) CLOSE IPS FILE, CLOSE COMTAL
C
0016        CALL IPCLOS
C
0017        IER = ICMCLO
C
```

```
RT-11 FORTRAN IV        V01B-08A   WED 18-APR-79 18:12:54       PAGE 002
      C
0018        RETURN
0019        END
```



```
RT-11 FORTRAN IV        STORAGE MAP

NAME    OFFSET  ATTRIBUTES

NFTYP   000014  INTEGER*2 ARRAY (2)
NAFTYP  000020  INTEGER*2 ARRAY (2)
IMNO    000066  INTEGER*2 VARIABLE
NRECS   000070  INTEGER*2 VARIABLE
NRECSZ  000072  INTEGER*2 VARIABLE
IER     000074  INTEGER*2 VARIABLE
IPWOPE  000000  INTEGER*2 PROCEDURE
ICMOPE  000000  INTEGER*2 PROCEDURE
IRFM    000000  INTEGER*2 PROCEDURE
IPUTFM  000000  INTEGER*2 PROCEDURE
IPCLOS  000000  INTEGER*2 PROCEDURE
ICMCLO  000076  INTEGER*2 VARIABLE

COMMON BLOCK /LINE/     LENGTH 002000

IBUFF   000000  INTEGER*2 ARRAY (512)
```

```
RT-11 FORTRAN IV        V01B-08A   WED 18-APR-79 18:12:57       PAGE 001
```

```
0001          C
              C     SUBROUTINE IWCBF
              C
              C     ,    THIS ROUTINE WILL SEND ONE LINE OF TEXT
              C     ,    TO THE COMTAL COMMAND BUFFER
              C
              C     ,    1) QUERY USER FOR TEXT
              C     ,    2) OPEN COMTAL
              C     ,    3) SEND TO COMTAL
              C     ,    4) CLOSE COMTAL
              C
              C
0002          C     LOGICAL*1 ITEXT(132)
0003          C     COMMON /LINE/ ITEXT
              C
              C
              C
              C     ,    1) QUERY USER
              C
0004          C     WRITE(7,100)
0005          100   FORMAT(' ENTER ONE LINE OF COMMAND TEXT, TERMINATE WITH',
                   $        ' $, CR. ',/,)
              C
0006          C     READ (7,200)L,(ITEXT(J),J=1,L)
0007          200   FORMAT(Q,130A1)
              C
              C
              C     ,    2) OPEN COMTAL
              C
0008          C     IER=ICMOPE()
              C
              C
              C     ,    3) SEND TO COMTAL
              C
0009          C     IER = IWCB(ITEXT)
              C
              C
              C     ,    4) CLOSE COMTAL
              C
0010          C     IER = ICMCLO
              C
0011          C     RETURN
0012          C     END
```

RT-11 FORTRAN IV STORAGE MAP

NAME OFFSET ATTRIBUTES

```
L         000120  INTEGER*2 VARIABLE
J         000122  INTEGER*2 VARIABLE
IER       000124  INTEGER*2 VARIABLE
ICMOPE    000000  INTEGER*2 PROCEDURE
IWCE      000000  INTEGER*2 PROCEDURE
ICMCLO    000126  INTEGER*2 VARIABLE

COMMON BLOCK /LINE/      LENGTH 000204

ITEXT     000000  LOGICAL*1 ARRAY (132)
&
```

```
C
C     ,                    FILE NAME: CPIPLB.FOR
C     ,                    DATE LAST EDITED: 21-MAR-79
C     ,
C     ,                    HISTORY:
C     ,                    DATE:           MODIFICATION
C     ,                    -----           ------------
C     ,                     6-MAR-79       INITAL WRITING
C     ,                     7-MAR-79       ADD MORE ROUTINES
C     ,                     8-MAR-79       FIX BUGS
C     ,                     9-MAR-79       FIX MORE BUGS
C     ,                    19-MAR-79       CHANGE SUBS TO FUNCTIONS
C     ,                    21-MAR-79       ADD MORE FUNCTIONS
C
C     ,    THIS FILE CONTAINS THE GENERAL SUBROUTINES
C     ,    FOR THE COMTAL READ AND WRITE IPS FILES PROGRAM.
C
C
0001       INTEGER FUNCTION IPROPE(IRECS,IRECSZ,IFTYP)
C
C
C     ,    THIS ROUTINE OPENS AN IPS FILE FOR READING.
C     ,
C     ,            IREC   = NUMBER OF RECORDS IN FILE
C     ,            IRECSZ = NUMBER OF WORDS PER RECORD
C     ,            IFTYP  = RAD50 FILE TYPE
C
0002       INTEGER     IFTYP(2)
C
C      DATA DEFINITION
C
0003       INTEGER INFL(4),OUTFL(4),EXT(2),FSPEC(39),BUFF(256,2)
C
0004       EQUIVALENCE (INFL,FSPEC(16)),(OUTFL,FSPEC(16))
C
C
0005       COMMON /IFILES/ FSPEC,EXT,NRECS,NRECSZ,ICHIPS,
          &                IACT,NABREC,BUFF
C
```

```
                C
0006                  DATA IACT / 1 /
0007                  DATA      EXT /3RIMG,3RIMG/
                C
                C     END OF DATA DEFINITION
                C
                C*****************************************************
                C
                C
                C     ,    1) QUERY USER FOR INPUT FILE NAME
```



```
                C     ,    2) GET CHANNEL
                C     ,    3) OPEN FILE
                C     ,    4) READ HEADER BLOCK, FILL IN PARAMETERS
                C
                C
                C     ,    1) QUERY USER
                C
0008                  IPROPE = 0
                C
                C     PAUSE '  IPROPE 1, BEGIN '
                C
0009  90        WRITE(7,100)
0010  100       FORMAT(' ENTER INPUT FILE NAME ',/)
                C
0011              IF(ICSI(FSPEC,IFTYP,,,0).NE.0)    GO TO 90
                C
                C     WRITE(7,124)
                C 124 FORMAT(/,/,'  FSPEC = ',/)
                C
                C     WRITE(7,123)FSPEC
                C 123 FORMAT(' ',1007)
                C
                C
                C     ,    2) GET CHANNEL
                C
                C     PAUSE 'IPROPE 3 JUST PRIOR TO GETC '
                C
0013              ICHIPS=IGETC()
0014              IF(ICHIPS.LT.0) STOP 'BAD CHANNEL'
                C
                C     ,    3) OPEN FILE
                C
```

```
C
C     PAUSE ' IPROPE 4 JUST PRIOR TO LOOKUP '
C
      INSIZ=LOOKUP(ICHIPS,INFL)
      IF(INSIZ.LT.0) GO TO 1100
C
C     ,    4) READ HEADER
C
C     PAUSE ' IPROPE 5 JUST PRIOR TO READW '
C
      NABREC=0
C
      IC=IREADW(256,BUFF(1,1),NABREC,ICHIPS)
      IF(IC.LT.0) GO TO 1100
C
C     ,    4.5) FILL HEADER INFORMATION
```



```
C
      NRECS  = BUFF(7,1)
      NRECSZ = BUFF(4,1)
C
C     ,            CHANGE RECORD SIZE FROM BYTES TO WORDS
C
      NRECSZ = (NRECSZ+1) / 2
C
      IRECS  = NRECS
      IRECSZ = NRECSZ
C
C     ,            SET FLAG BUFFER EMPTY
C
      IACT = 1
      NABREC = NABREC + 1
      RETURN
C
C     ,            ERROR OCCURED
C
1100  WRITE(7,110)
110   FORMAT(' ERROR , IPROPE  ')
      IPROPE = -1
      RETURN
C
      END
```

```
RT-11 FORTRAN IV        STORAGE MAP

NAME    OFFSET  ATTRIBUTES

IFTYP   000020  INTEGER*2 PARAMETER ARRAY (2)
IFROPE  000022  INTEGER*2 VARIABLE
IRECS   000014  INTEGER*2 PARAMETER VARIABLE
IRECSZ  000016  INTEGER*2 PARAMETER VARIABLE
ICSI    000000  INTEGER*2 PROCEDURE
IGETC   000000  INTEGER*2 PROCEDURE
INSIC   000132  INTEGER*2 VARIABLE
LOOKUP  000000  INTEGER*2 PROCEDURE
IC      000134  INTEGER*2 VARIABLE
IREADW  000000  INTEGER*2 PROCEDURE

COMMON BLOCK /IFILES/   LENGTH 002134

FSPEC   000000  INTEGER*2 ARRAY (39)
EXT     000116  INTEGER*2 ARRAY (2)
NRECS   000122  INTEGER*2 VARIABLE
NRECSZ  000124  INTEGER*2 VARIABLE
ICHIPS  000126  INTEGER*2 VARIABLE
IACT    000130  INTEGER*2 VARIABLE
NABREC  000132  INTEGER*2 VARIABLE
BUFF    000134  INTEGER*2 ARRAY (256,2) VECTORED
INFL    000036  INTEGER*2 ARRAY (4)
OUTFL   000036  INTEGER*2 ARRAY (4)
```



```
RT-11 FORTRAN IV        V01B-08A   WED 21-MAR-79 17:21:11       PAGE 001

      C
0001        INTEGER FUNCTION IPWOPE(IRECS,IRECSZ,IFTYP,IAFTYP)
      C
      C
      C     .    THIS ROUTINE OPENS AN IPS FILE FOR OUTPUT.
      C
      C
      C     ,            IRECS  =  NUMBER OF RECORDS IN FILE
      C     ,            IRECSZ =  RECORD SIZE IN WORDS
      C     ,            IFTYP  =  RAD50 FILE TYPE
      C     ,            IAFTYP =  ASCII FILE TYPE
      C
0002        INTEGER IFTYP(2),IAFTYP(2)
      C
      C
```

```
      C     DATA DEFINITION
      C
0003        INTEGER AEXT(2)
      C
      C
0004        INTEGER INFL(4),OUTFL(4),EXT(2),BUFF(256,2)
0005        INTEGER     FSPEC(39)
      C
0006        EQUIVALENCE (INFL,FSPEC(16)),(OUTFL,FSPEC(16))
      C
      C
0007        COMMON /IFILES/ FSPEC,EXT,NRECS,NRECSZ,ICHIPS,
     &                        IACT,NABREC,BUFF
      C
      C
0008        DATA AEXT / '.I', 'MG' /
      C
      C     END OF DATA DEFINITION
      C
      C**********************************************
      C
      C
      C
      C     ,    1) SET UP BUFFR FOR HEADER FILE
      C     ,    2) GET CHANNEL
      C     ,    3) QUERY USER FOR FILE
      C     ,    4) OPEN FILE
      C     ,    5) WRITE HEADER
      C
      C
      C
      C
      C     ,    1)SET UP HEADER
      C
      C
      C     PAUSE ' IWOPE 1 BEGIN '
      C
0009        IPWOPE = 0
      C
0010        DO 10 J=1,256
      C
```



```
0011        BUFF(J,1) = 0
      C
0012  10    CONTINUE
      C
      C
      C     ,          NOTE: I IGNOR RECORD SIZE, ALL COMTAL FILES
      C     ,          WILL BE 256 BY 256
```

```
C
0013        BUFF(7,1) = IRECS
0014        BUFF(4,1) = IRECSZ
C
0015        BUFF(2,1) = IAFTYP(1)
0016        BUFF(3,1) = IAFTYP(2)
C
C           SINCE NUMBER OF RECORDS = 256 ,
C           NUMBER OF BLOCKS = 256 * (256/512) + 1 , = 129
C
0017        ARECS=IRECS
0018        ARECSZ=IRECSZ
C
0019        ASIZ = ((ARECS*ARECSZ)/256.) + 2.
C
0020        BUFF(5,1) = ASIZ
C
C
C     ,     2) GET CHANNEL
C
0021        ICHIPS=IGETC()
0022        IF(ICHIPS.LT.0) STOP 'BAD CHANNEL'
C
C
C     ,     3) QUERY USER
C
0024  90    WRITE(7,100)
0025  100   FORMAT(' ENTER OUTPUT FILE NAME ',/)
C
0026        IF(ICSI(FSPEC,IFTYP,,,0).NE.0)    GO TO 90
C
C
C     ,     4) OPEN FILE
C
0028        IF(IENTER(ICHIPS,OUTFL,0).LT.0) GO TO 1100
C
C
C     ,     5) WRITE HEADER
C
0030        NABREC=0
C
0031        IC= IWRITW(256,BUFF(1,1),NABREC,ICHIPS)
C
C
0032        NABREC = NABREC+1
0033        RETURN
```

```
C
C
C
C     ,    ERROR OCCURED
C
0034  1100  WRITE(7,110)
0035  110   FORMAT(' ERROR , IPUOPE  ')
C
0036        IPUOPE = -1
0037        RETURN
0038        END
```


NAME	OFFSET	ATTRIBUTES
IFTYP	000020	INTEGER*2 PARAMETER ARRAY (2)
IWFTYP	000022	INTEGER*2 PARAMETER ARRAY (2)
AEXT	000024	INTEGER*2 ARRAY (2)
IPUOPE	000030	INTEGER*2 VARIABLE
IRECS	000014	INTEGER*2 PARAMETER VARIABLE
IRECSZ	000016	INTEGER*2 PARAMETER VARIABLE
J	000142	INTEGER*2 VARIABLE
APECS	000144	REAL*4 VARIABLE
ARECSZ	000150	REAL*4 VARIABLE
ASIZ	000154	REAL*4 VARIABLE
IGETC	000000	INTEGER*2 PROCEDURE
IGSI	000000	INTEGER*2 PROCEDURE
IENTER	000000	INTEGER*2 PROCEDURE
IC	000160	INTEGER*2 VARIABLE
IWRITW	000000	INTEGER*2 PROCEDURE

COMMON BLOCK /IFILES/ LENGTH 002134

FSPEC	000000	INTEGER*2 ARRAY (39)
EXT	000116	INTEGER*2 ARRAY (2)
NRECS	000122	INTEGER*2 VARIABLE
NRECSZ	000124	INTEGER*2 VARIABLE
ICHIPS	000126	INTEGER*2 VARIABLE
IWCT	000130	INTEGER*2 VARIABLE
NWBPEC	000132	INTEGER*2 VARIABLE
BUFF	000134	INTEGER*2 ARRAY (256,2) VECTORED
INFL	000036	INTEGER*2 ARRAY (4)
OUTFL	000036	INTEGER*2 ARRAY (4)

```
C
0001      INTEGER FUNCTION IPRDIM(IBUFF)
C
C     ,    THIS ROUTINE WILL READ ONE RECORD FROM AN IPS FILE
C     ,    AND RETURN THE RECORD TO THE USER BUFFER.  THE IPS
C     ,    FILE MUST HAVE BEEN ALREADY OPENED BY A CALL TO
C     ,    SUBROUTINE  ' IPROPE ' .
C
C     ,    PARAMETERS:
C     ,            IBUFF  = INTEGER ARRAY FOR THE RECORD
C
C*****************************************************************
C     DATA DEFINITION
C     ,,,,,,,,,,,,,,,,
C     THIS PART OF THE DATA DEF. APPLIES ONLY TO THIS ROUTINE
C
0002      INTEGER  RECST,RECPOS
C
C     ,    <--------- SWAP ----------
C
0003      LOGICAL*1 IBUFF(1),BUFF(512,2)
C
C     ,    NOTE: IBUFF SHOULD BE MADE INTEGER TO SPEED UP DO
C     ,    LOOPS, CANT UNTIL BYTE SWAP FIXED
C
C     ,    <--------- SWAP ---------
C     ,,,,,,,,,,,,,,,,
C
0004      INTEGER INFL(4),OUTFL(4),EXT(2),FSPEC(39)
0005      EQUIVALENCE (INFL,FSPEC(16)),(OUTFL,FSPEC(16))
C
C       .
C
C
C
0006      COMMON /IFILES/ FSPEC,EXT,NRECS,NRECSZ,ICHIPS,
     *                    IACT,NABREC,BUFF
C
C     ,    <--------- SWAP ---------
C
0007      DATA IBFSZ /256 /
```

```
C     ,    <--------- SWAP ---------
C
C
C
```

```
0010  100   IACT=2
C
C     ,            READ 2 BLOCKS INTO THE BUFFERS
C
0011         IC=IREADW(512,BUFF(1,1),NABREC,ICHIPS)
C
C     ,            INCREMENT ABSOLUTE BLOCK NUMBER
C     ,            AND THEN READ RECORD
C
0012         NABREC = NABREC+2
0013         RECST=0
0014         GO TO 10
      C
```



```
      C
C     ,    2.2) READ RECORD INTO USER BUFFER - ACT BUFFER = 2
C     ,         (IACT = 2, BUT FIRST BUFFER)
C
C
C
0015  200   DO 2000 J=1,NRECSZ
      C
0016         RECPOS =RECST+J
0017         IF(RECPOS.GT.IBFSZ) GO TO 2100
      C
      C     ,    <--------- SWAP ---------
0019         NJ =2*J
0020         MJ =2*(RECPOS)
      C
0021         IBUFF(NJ-1) = BUFF(MJ,1)
0022         IBUFF(NJ)   = BUFF(MJ-1,1)
      C
C     ,    <--------- SWAP ---------
C
0023  2000   CONTINUE
      C
0024         RECST =RECST + NRECSZ
0025         GO TO 5000
      C
C
C     ,            END OF BUFFER BEFORE END OF RECORD
C     ,            SWITCH BUFFERS, FINISH READIND RECORD
C
0026  2100   IACT=3
0027         RECPOS=0
      C
      C
```

```
C
0025          DO 2300 M=J,NRECSZ
0029          RECPOS=RECPOS + 1
C
C
C             ,    <--------- SWAP ---------
0030          NJ = 2*M
0031          MJ =2*(RECPOS)
C
0032          IBUFF(NJ-1) = BUFF(MJ,2)
0033          IBUFF(NJ)   = BUFF(MJ-1,2)
C
C             ,    <--------- SWAP ---------
C
0034     2300 CONTINUE
0035          RECST=RECPOS
C
C
C             ,    4.1) READ BLOCK INTO EMPTY FIRST BUFFER
C
0036          IC=IREADW(256,BUFF(1,1),NABREC,ICHIPS)
C
```



```
0037          NABREC=NABREC+1
0038          GO TO 5000
C
C
C
C             ,    2.3) READ RECORD INTO USER BUFFER - ACTIVE BUFFER = 3
C
0039     3000 DO 3000     J=1,NRECSZ
C
0040          RECPOS =RECST + J
0041          IF(RECPOS.GT.IBFSZ) GO TO 3100
C
C
C             ,    <--------- SWAP ---------
0043          NJ = 2*J
0044          MJ =2*(RECPOS)
C
0045          IBUFF(NJ-1) = BUFF(MJ,2)
0046          IBUFF(NJ)   = BUFF(MJ-1,2)
C
C             ,    <--------- SWAP ---------
C
0047     3000 CONTINUE
C
0048          RECST=RECST + NRECSZ
0049          GO TO 5000
```

```
C
C
C
C     ;            END OF BUFFER BEFORE END OF RECORD
C     ;            SWITCH BUFFERS, FINISH READIND RECORD
C
0050  3100  IACT=2
0051        RECPOS=0
C
0052        DO 3300     M=J,NRECSZ
0053        RECPOS=RECPOS + 1
C
C     ,     <--------- SWAP ---------
0054        NJ = 2*M
0055        MJ =2*(RECPOS)
C
0056        IBUFF(NJ-1) = BUFF(MJ,1)
0057        IBUFF(NJ)   = BUFF(MJ-1,1)
C
C     ,     <--------- SWAP ---------
C
0058  3300  CONTINUE
0059        RECST=RECPOS
C
C     ,     4.1) READ BLOCK INTO EMPTY SECOND BUFFER
C
0060        IC=IREADW(256,BUFF(1,2),NABREC,ICHIPS)
C
0061        NABREC=NABREC+1
```



```
0062        GO TO 5000
C
C
C
C     ,     FINISH USER BUFFER WITH ZEROS
C
0063  5000  CONTINUE
C
C     ,     ''''''
```

```
C
C     ,    ''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
C     ,    <--------- SWAP -----------
C
0064          IF(NRECSZ.GE.256) GO TO 5200
C
0066          DO 5100 M=NRECSZ+1,256
C
C     ,    <--------- SWAP -----------
C
0067          M2 =M*2
0068          IBUFF(M2) = 0
C
0069          IBUFF(M2-1)=0
C
C     ,    <--------- SWAP -----------
C
0070    5100  CONTINUE
0071    5200  CONTINUE
C
0072          RETURN
0073          END
```



```
PT-11 FORTRAN IV        STORAGE MAP

NAME    OFFSET  ATTRIBUTES

IBUFF   000014  LOGICAL*1 PARAMETER ARRAY (1)
IPRDIM  000016  INTEGER*2 VARIABLE
RECST   000032  INTEGER*2 VARIABLE
RECPOS  000034  INTEGER*2 VARIABLE
IBFSZ   000020  INTEGER*2 VARIABLE
IC      000036  INTEGER*2 VARIABLE
IPEADU  000000  INTEGER*2 PROCEDURE
J       000040  INTEGER*2 VARIABLE
NJ      000042  INTEGER*2 VARIABLE
MJ      000044  INTEGER*2 VARIABLE
M       000046  INTEGER*2 VARIABLE
M2      000050  INTEGER*2 VARIABLE

COMMON BLOCK /IFILES/   LENGTH 002134

FSPEC   000000  INTEGER*2 ARRAY (39)
EXT     000116  INTEGER*2 ARRAY (2)
NRECS   000122  INTEGER*2 VARIABLE
NRECSZ  000124  INTEGER*2 VARIABLE
ICHIPS  000126  INTEGER*2 VARIABLE
```

```
IWCT    000130  INTEGER*2 VARIABLE
NWEREC  000132  INTEGER*2 VARIABLE
BUFF    000134  LOGICAL*1 ARRAY (512,2) VECTORED
INFL    000036  INTEGER*2 ARRAY (4)
OUTFL   000036  INTEGER*2 ARRAY (4)



RT-11 FORTRAN IV        V01B-08A   WED 21-MAR-79 17:21:19       PAGE 001

      C
      C
0001          INTEGER FUNCTION IPUTIM (IBUFF)
      C
      C
      C
      C     ,    THIS ROUTINE WILL WRITE ONE RECORD TO AN IPS FILE
      C     ,    FROM THE RECORD IN THE USER BUFFER.  THE IPS
      C     ,    FILE MUST HAVE BEEN ALREADY OPENED BY A CALL TO
      C     ,    SUBROUTINE  ' IPUOPE ' .
      C
      C     ,    PARAMETERS:
      C
      C     ,            IBUFF  = INTEGER ARRAY FOR THE RECORD
      C
      C
      C     **********************************************************
      C
      C     DATA DEFINITION
      C     ,,,,,,,,,,,,,,,,,
      C
      C     THIS PART OF THE DATA DEF. APPLIES ONLY TO THIS ROUTINE
      C
      C     ,    <--------- SWAP ---------
      C
0002          LOGICAL*1 IBUFF(1), BUFF(256,4)
      C
      C     NOTE: BUFF IS CHANGED TO 128,4 FROM 256,2 FOR THIS ROUTINE
      C
      C     ,    <--------- SWAP ----------
      C
      C         .
      C
      C     ,,,,,,,,,,,,,,,,,
      C
      C
0003          INTEGER INFL(4),OUTFL(4),EXT(2),FSPEC(39)
      C
0004          EQUIVALENCE (INFL,FSPEC(16)),(OUTFL,FSPEC(16))
      C
      C
      C
```

```
C     ,    <--------- SWAP ----------

0005       COMMON /IFILES/ FSPEC,EXT,NRECS,NRECSZ,ICHIPS,
     *                     IACT,NABREC,BUFF

C     ,,,,,,,,,,,,,,,,,,,
      THIS PART OF THE DATA DEF. APPLIES ONLY TO THIS ROUTINE
0006       DATA N / 1 /
```



```
C     ,,,,,,,,,,,,,,,,,,,,,,
      END OF DATA DEFINITION
C*****************************************************

C     ,    1) FILL BUFF WITH USER DATA
C     ,    2) UPON 4 DATA RECORDS, WRITE ONE BLOCK


C     ,    1) FILL BUFFER WITH DATA

      PAUSE ' IPWITE 1 BEGIN '

C     ,    ,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,
C     ,    NOTE:
C     ,    ----
C     ,              THE CODE MARKED  <-----SWAP---
C     ,              IS FOR THE BYTE SWAP THAT THE COMTAL
C     ,              IS SUPPOSED TO DO, BUT AT THIS TIME
C     ,              DOESN'T.
C     ,    ,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,
0007       IPUTIM = 0
```

```
0008          DO 100      J=1,128
              C
              C     ,    <--------- SWAP -----------
              C
0009          MJ = J*2
              C
              C
              C
              C     ,    <--------- SWAP -----------
              C
              C
0010          BUFF(MJ,N)   = IBUFF(MJ-1)
0011          BUFF(MJ-1,N) = IBUFF(MJ)
              C
0012  100     CONTINUE
              C
0013          N = N+ 1
0014          IF(N.GT.4) GO TO 1000
              C
0016          RETURN
              C
              C
              C     ,    2) WRITE OUT BUFFER
              C
```



```
0017  1000  N =1
              C
0018          IC = IUWRITW(512,BUFF(1,1),NABREC,ICHIPS)
              C
0019          NABREC = NABREC + 2
0020          RETURN
0021          END
```

PT-11 FORTRAN IV   STORAGE MAP

NAME	OFFSET	ATTRIBUTES
IBUFF	000014	LOGICAL*1 PARAMETER ARRAY (1)
IPUTIM	000016	INTEGER*2 VARIABLE
N	000020	INTEGER*2 VARIABLE
J	000034	INTEGER*2 VARIABLE
MJ	000036	INTEGER*2 VARIABLE
IC	000040	INTEGER*2 VARIABLE

```
IURITU  000000  INTEGER*2 PROCEDURE

COMMON BLOCK  IFILES/   LENGTH 002134

FSPEC    000000  INTEGER*2 ARRAY (39)
EXT      000116  INTEGER*2 ARRAY (2)
NRECS    000122  INTEGER*2 VARIABLE
NRECSZ   000124  INTEGER*2 VARIABLE
ICHIPS   000126  INTEGER*2 VARIABLE
IACT     000130  INTEGER*2 VARIABLE
NABREC   000132  INTEGER*2 VARIABLE
BUFF     000134  LOGICAL*1 ARRAY (256,4) VECTORED
INFL     000036  INTEGER*2 ARRAY (4)
OUTFL    000036  INTEGER*2 ARRAY (4)
```



```
0001      C
          C     SUBROUTINE IPCLOS
          C
          C     ,    THIS ROUTINE WILL CLOSE THE IPS FILE
          C
          C
          C     DATA DEFINITION
          C
0002      C     INTEGER INFL(4),OUTFL(4),EXT(2),FSPEC(39),BUFF(256,2)
          C
0003      C     EQUIVALENCE (INFL,FSPEC(16)),(OUTFL,FSPEC(16))
          C
0004      C     COMMON /IFILES/ FSPEC,EXT,NRECS,NRECSZ,ICHIPS,
          *                     IACT,NABREC,BUFF
          C
          C
          C     END OF DATA DEFINITION
          C
          C*************************************************
          C
          C
0005      C     CALL CLOSEC(ICHIPS)
          C
0006      C     RETURN
0007      C     END
```

```
NAME    OFFSET  ATTRIBUTES

CLOSEC  000000  REAL*4    PROCEDURE

COMMON BLOCK /IFILES/   LENGTH 002134

FSPEC   000000  INTEGER*2 ARRAY (39)
EXT     000116  INTEGER*2 ARRAY (2)
NRECS   000122  INTEGER*2 VARIABLE
NRECSZ  000124  INTEGER*2 VARIABLE
ICHIPS  000126  INTEGER*2 VARIABLE
IACT    000130  INTEGER*2 VARIABLE
NABREC  000132  INTEGER*2 VARIABLE
BUFF    000134  INTEGER*2 ARRAY (256,2) VECTORED
INFL    000036  INTEGER*2 ARRAY (4)
OUTFL   000036  INTEGER*2 ARRAY (4)
```

```
RT-11 FORTRAN IV        V01B-08A   WED 21-MAR-79 17:21:24       PAGE 001

0001            C
                      INTEGER FUNCTION IPRDPS (IBUFF)
                C
                C     ,    FUNCTIONAL DESCRIPTION:
                C     ,
                C     ,    THIS ROUTINE READS ONE PSEUDO COLOR MEMORY
                C     ,    RECORD INTO THE USER BUFFER
                C     ,
                C     ,    THIS APPLIES ONLY TO THIS ROUTINE:
                C
0002                  INTEGER IBUFF(1)
                C     , ***************************************
                C
                      DATA DEFINITION
0003                  INTEGER INFL(4),OUTFL(4),EXT(2),FSPEC(39),BUFF(256,2)
0004                  EQUIVALENCE (INFL,FSPEC(16)),(OUTFL,FSPEC(16))
0005                  COMMON /IFILES/ FSPEC,EXT,NRECS,NRECSZ,ICHIPS,
                     *                IACT,NABREC,BUFF
                C
                      E D OF DATA DEFINITION
                C***************************************************
                C
```

```
      C
0006          IPRDPS = 0
      C
0007          NSIZ = 256
0008          NR = 1
      C
0009          IC = IREADW(NSIZ,IBUFF,NR,ICHIPS)
0010          IF(IC.LT.0) IPRDPS = IC
      C
0012          RETURN
0013          END
```

```
RT-11 FORTRAN IV        STORAGE MAP

NAME    OFFSET  ATTRIBUTES

IBUFF   000014  INTEGER*2 PARAMETER ARRAY (1)
IPRDPS  000016  INTEGER*2 VARIABLE
NSIZ    000024  INTEGER*2 VARIABLE
NR      000026  INTEGER*2 VARIABLE
IC      000030  INTEGER*2 VARIABLE
IREADW  000000  INTEGER*2 PROCEDURE

COMMON BLOCK /IFILES/   LENGTH 002134

FSPEC   000000  INTEGER*2 ARRAY (39)
EXT     000116  INTEGER*2 ARRAY (2)
NRECS   000122  INTEGER*2 VARIABLE
NRECSZ  000124  INTEGER*2 VARIABLE
ICHIPS  000126  INTEGER*2 VARIABLE
IHOT    000130  INTEGER*2 VARIABLE
NAMREC  000132  INTEGER*2 VARIABLE
BUFF    000134  INTEGER*2 ARRAY (256,2) VECTORED
INFL    000036  INTEGER*2 ARRAY (4)
OUTFL   000036  INTEGER*2 ARRAY (4)
```

```
RT-11 FORTRAN IV        V01B-08A   WED 21-MAR-79 17:21:27       PAGE 001

      C
0001          INTEGER FUNCTION IPWTPS (IBUFF)
      C
      C     ,    FUNCTIONAL DESCRIPTION:
      C
      C     ,    THIS ROUTINE WRITES THE PSEUDO COLOR MEMORY
```

```
C     ,    RECORD TO THE IPS FILE.
C     ,
C     ,    THIS APPLIES ONLY TO THIS ROUTINE:
C
0002        INTEGER IBUFF(1)
C     , ***************************************
C
C     DATA DEFINITION
C
0003        INTEGER INFL(4),OUTFL(4),EXT(2),FSPEC(39),BUFF(256,2)
C
0004        EQUIVALENCE (INFL,FSPEC(16)),(OUTFL,FSPEC(16))
C
0005        COMMON /IFILES/ FSPEC,EXT,NRECS,NRECSZ,ICHIPS,
           $                IACT,NABREC,BUFF
C
C     END OF DATA DEFINITION
C*************************************************
C
C
0006        IPUTPS = 0
C
0007        NR=1
0008        NSIZ=256
C
0009        IC = IURITW(NSIZ,IBUFF,NR,ICHIPS)
0010        IF(IC.LT.0) IPUTPS = IC
C
0012        RETURN
0013        END
```



```
RT-11 FORTRAN IV        STORAGE MAP

NAME    OFFSET  ATTRIBUTES

IBUFF   000014  INTEGER*2 PARAMETER ARRAY (1)
IPUTPS  000016  INTEGER*2 VARIABLE
NR      000024  INTEGER*2 VARIABLE
NSIZ    000026  INTEGER*2 VARIABLE
IC      000030  INTEGER*2 VARIABLE
IURITW  000000  INTEGER*2 PROCEDURE

COMMON BLOCK /IFILES/   LENGTH 002134

FSPEC   000000  INTEGER*2 ARRAY (39)
```

```
EXT      000116  INTEGER*2 ARRAY (2)
NRECS    000122  INTEGER*2 VARIABLE
NRECSZ   000124  INTEGER*2 VARIABLE
ICHIPS   000126  INTEGER*2 VARIABLE
IACT     000130  INTEGER*2 VARIABLE
NABREC   000132  INTEGER*2 VARIABLE
BUFF     000134  INTEGER*2 ARRAY (256,2) VECTORED
INFL     000036  INTEGER*2 ARRAY (4)
OUTFL    000036  INTEGER*2 ARRAY (4)
```

```
RT-11 FORTRAN IV        V01B-08A   WED 21-MAR-79 17:21:30       PAGE 001

0001            C
                      INTEGER FUNCTION IPRDGM (IBUFF)
                C
                C     ,    FUNCTIONAL DESCRIPTION:
                C     ,
                C     ,    THIS ROUTINE READS ONE LINE OF GRAPHICS MEMORY
                C     ,    RECORD INTO THE USER BUFFER
                C     ,
                C     ,    THIS APPLIES ONLY TO THIS ROUTINE:
                C
0002                  INTEGER IBUFF(1)
                C
                C     NOTE: BUFF IS DEFINED AS 16 BY 16 FOR 16 ROWS
                C    16 WORDS (2 BYTES) OF GRAPHICS MEMORY PER BLOCK
                C     THIS DIFFERS FROM THE OTHERS ROUTINES
0003                  INTEGER BUFF(16,16)
                C     , ***************************************
                C
                C     DATA DEFINITION
                C
0004                  INTEGER INFL(4),OUTFL(4),EXT(2),FSPEC(39)
0005            C     EQUIVALENCE (INFL,FSPEC(16)),(OUTFL,FSPEC(16))
0006            C     COMMON /IFILES/ FSPEC,EXT,NRECS,NRECSZ,ICHIPS,
                     *                IACT,NABREC,BUFF
0007            C     DATA IRELRC /1/
                C
                C     END OF DATA DEFINITION
                C*****************************************************
                C
```

```
0008          IPRDGM = 0
              C
0009          NSIZ = 256
              C
              C     1) CHECK FOR FIRST CALL - NEED TO READ IN BLOCK
              C
0010          IF(IACT.EQ.1) GO TO 100
              C
              C     2) CHECK FOR BLOCK EMPTY - NEED TO READ IN BLOCK
              C
0012    200   IF(IRELRC.GE.17) GO TO 100
              C
              C     3) MOVE A RECORD TO USER BUFFER
              C
0014          DO 300 J=1,16
```



```
              C
0015          IBUFF(J) = BUFF(J,IRELRC)
              C
0016    300   CONTINUE
              C
              C     4) INCREMENT RELATIVE RECORD NUMBER
              C
0017          IRELRC =IRELRC + 1
              C
0018          RETURN
              C
              C     5) END OF BLOCK READ IN A NEW BLOCK
              C
0019    100   IACT = 2
              C
0020          IC = IREADW(NSIZ,BUFF(1,1),NABREC,ICHIPS)
0021          IF(IC.LT.0) IPRDGM = IC
              C
0023          IRELRC = 1
0024          NABREC = NABREC + 1
0025          GO TO 200
              C
0026          END
```

```
RT-11 FORTRAN IV        STORAGE MAP

NAME    OFFSET  ATTRIBUTES

IBUFF   000014  INTEGER*2 PARAMETER ARRAY (1)
IPRDGM  000016  INTEGER*2 VARIABLE
IRELRC  000020  INTEGER*2 VARIABLE
NSIZ    000062  INTEGER*2 VARIABLE
J       000064  INTEGER*2 VARIABLE
IC      000066  INTEGER*2 VARIABLE
IREADW  000000  INTEGER*2 PROCEDURE

COMMON BLOCK /IFILES/   LENGTH 001134

FSPEC   000000  INTEGER*2 ARRAY (39)
EXT     000116  INTEGER*2 ARRAY (2)
NRECS   000122  INTEGER*2 VARIABLE
NRECSZ  000124  INTEGER*2 VARIABLE
ICHIPS  000126  INTEGER*2 VARIABLE
IACT    000130  INTEGER*2 VARIABLE
NABREC  000132  INTEGER*2 VARIABLE
BUFF    000134  INTEGER*2 ARRAY (16,16) VECTORED
INFL    000036  INTEGER*2 ARRAY (4)
OUTFL   000036  INTEGER*2 ARRAY (4)
```



```
RT-11 FORTRAN IV        V01B-08A   WED 21-MAR-79 17:21:33       PAGE 001

        C
0001          INTEGER FUNCTION IPUTGM (IBUFF)
        C
        C     ,    FUNCTIONAL DESCRIPTION:
        C
        C     ,    THIS ROUTINE WRITES THE GRAPHICS MEMORY
        C     ,    RECORD TO THE IPS FILE.
        C
        C
        C     ,    THIS APPLIES ONLY TO THIS ROUTINE:
        C
        C
0002          INTEGER IBUFF(1)
        C
        C
        C     NOTE: BUFF IS DEFINED AS 16 BY 16 FOR 16 ROWS
        C    16 WORDS (2 BYTES) OF GRAPHICS MEMORY PER BLOCK
        C     THIS DIFFERS FROM THE OTHERS ROUTINES
        C
0003          INTEGER BUFF(16,16)
        C
```

```
C     , ***************************************
C
C     DATA DEFINITION
C
0004        INTEGER INFL(4),OUTFL(4),EXT(2),FSPEC(39)
C
0005        EQUIVALENCE (INFL,FSPEC(16)),(OUTFL,FSPEC(16))
C
0006        COMMON /IFILES/ FSPEC,EXT,NRECS,NRECSZ,ICHIPS,
           *                IACT,NABREC,BUFF
C
0007        DATA IRELRC /1/
C
C     END OF DATA DEFINITION
C
C***********************************************
C
C
0008        IPUTGM = 0
C
0009        NSIZ=256
C
0010        DO 100      J=1,16
C
0011        BUFF(J,IRELRC) = IBUFF(J)
C
0012  100   CONTINUE
C
0013        IRELRC=IRELRC + 1
C
0014        IF(IRELRC.GE.17) GO TO 1000
C
0016        RETURN
C
C
```



```
C
0017  1000  IRELRC = 1
C
0018        IC = IWRITW(NSIZ,BUFF(1,1),NABREC,ICHIPS)
0019        IF(IC.LT.0) IPUTGM = IC
0021        NABREC =NABREC + 1
C
0022        RETURN
0023        END
```

```
RT-11 FORTRAN IV          STORAGE MAP

NAME     OFFSET  ATTRIBUTES

IBUFF    000014  INTEGER*2 PARAMETER ARRAY (1)
IFUTGM   000016  INTEGER*2 VARIABLE
IRELRC   000020  INTEGER*2 VARIABLE
NSIZ     000062  INTEGER*2 VARIABLE
J        000064  INTEGER*2 VARIABLE
IC       000066  INTEGER*2 VARIABLE
IWRITW   000000  INTEGER*2 PROCEDURE

COMMON BLOCK /IFILES/   LENGTH 001134

FSPEC    000000  INTEGER*2 ARRAY (39)
EXT      000116  INTEGER*2 ARRAY (2)
NRECS    000122  INTEGER*2 VARIABLE
NRECS2   000124  INTEGER*2 VARIABLE
ICHIPS   000126  INTEGER*2 VARIABLE
IACT     000130  INTEGER*2 VARIABLE
NABREC   000132  INTEGER*2 VARIABLE
BUFF     000134  INTEGER*2 ARRAY (16,16) VECTORED
INFL     000036  INTEGER*2 ARRAY (4)
OUTFL    000036  INTEGER*2 ARRAY (4)
```



```
RT-11 FORTRAN IV        V01B-08A   WED 21-MAR-79 17:21:36        PAGE 001

0001      C
          INTEGER FUNCTION IPRDFM (IBUFF)
          C
          C     FUNCTIONAL DESCRIPTION:
          C
          C     THIS ROUTINE READS ONE FUNCTION MEMORY
          C     RECORD INTO THE USER BUFFER
          C
          C
          C     THIS APPLIES ONLY TO THIS ROUTINE:
          C
          C
0002      C      INTEGER IBUFF(1)
          C
          C     NOTE: BUFF IS DEFINED AS 512 BYTES BY 1
          C     THIS IS BECAUSE .TRA FILES ARE ONE VALUE
          C     PER BYTE WHEREAS THE COMTAL FUNCTION MEMORY
          C     IS ONE VALUE PER WORD (2 BYTES)
          C
```

```
0003          LOGICAL*1 BUFF(512,1)
              C
              C*************************************
              C
              C     DATA DEFINITION
              C
0004          INTEGER INFL(4),OUTFL(4),EXT(2),FSPEC(39)
              C
0005          EQUIVALENCE (INFL,FSPEC(16)),(OUTFL,FSPEC(16))
              C
0006          COMMON /IFILES/ FSPEC,EXT,NRECS,NRECSZ,ICHIPS,
             *                IACT,NABREC,BUFF
              C
              C     END OF DATA DEFINITION
              C
              C*************************************************
              C
0007          IPRDFM = 0
              C
0008          NSIZ = 256
0009          NR = 1
              C
0010          IC = IREADW(NSIZ,BUFF(1,1),NR,ICHIPS)
0011          IF(IC.LT.0) IPRDFM = IC
              C
              C     CONVERT PER BYTE TO PER WORD
              C
0013          DO 100 J = 1,256
              C
0014          IBUFF(J) = BUFF(J,1)
              C
              C     CORRECT FOR NEGATIVE
              C
```



```
0015          IF(IBUFF(J).LT.0) IBUFF(J) = IBUFF(J) + '400
              C
0017  100     CONTINUE
              C
0018          RETURN
0019          END
```

```
RT-11 FORTRAN IV        STORAGE MAP

NAME    OFFSET  ATTRIBUTES

IBUFF   000014  INTEGER*2 PARAMETER ARRAY (1)
IPRDFM  000016  INTEGER*2 VARIABLE
NSIZ    000022  INTEGER*2 VARIABLE
NR      000024  INTEGER*2 VARIABLE
IC      000026  INTEGER*2 VARIABLE
IREADW  000000  INTEGER*2 PROCEDURE
J       000030  INTEGER*2 VARIABLE

COMMON BLOCK /IFILES/   LENGTH 001134

FSPEC   000000  INTEGER*2 ARRAY (39)
EXT     000116  INTEGER*2 ARRAY (2)
NRECS   000122  INTEGER*2 VARIABLE
NRECSZ  000124  INTEGER*2 VARIABLE
ICHIPS  000126  INTEGER*2 VARIABLE
IACT    000130  INTEGER*2 VARIABLE
NABREC  000132  INTEGER*2 VARIABLE
BUFF    000134  LOGICAL*1 ARRAY (512,1) VECTORED
INFL    000036  INTEGER*2 ARRAY (4)
OUTFL   000036  INTEGER*2 ARRAY (4)
```



```
RT-11 FORTRAN IV        V01B-08A   WED 21-MAR-79 17:21:38        PAGE 001

0001            C
                      INTEGER FUNCTION IPUTFM (IBUFF)
                C
                C     FUNCTIONAL DESCRIPTION:
                C
                C     THIS ROUTINE WRITES THE FUNCTION MEMORY
                C     RECORD TO THE IPS FILE.
                C
                C     THIS APPLIES ONLY TO THIS ROUTINE:
                C
0002                  INTEGER IBUFF(1)
                C
                C     NOTE: BUFF IS DEFINED AS 512 BYTES BY 1
                C     THIS IS BECAUSE .TRA FILES ARE ONE VALUE
                C     PER BYTE WHEREAS THE CORTAL FUNCTION MEMORY
                C     IS ONE VALUE PER WORD (2 BYTES)
                C
```

```
0003          LOGICAL*1 BUFF(512,1)
C
C     , xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
C
C     DATA DEFINITION
C
0004          INTEGER INFL(4),OUTFL(4),EXT(2),FSPEC(39)
C
0005          EQUIVALENCE (INFL,FSPEC(16)),(OUTFL,FSPEC(16))
C
0006          COMMON /IFILES/ FSPEC,EXT,NRECS,NRECSZ,ICHIPS,
             &                    IACT,NABREC,BUFF
C
C     END OF DATA DEFINITION
C
C xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
C
0007          IPUTFM = 0
C
C     ,    1) CLEAR BUFFER
C
0008          DO 100      J=255,512
C
0009          BUFF(J,1) = 0
C
0010     100  CONTINUE
C
C     ,    2) PACK BLOCK
C
0011          DO 200      J=1,256
C
0012          BUFF(J,1) =IBUFF(J)
```



```
C
0013     200  CONTINUE
C
C     ,    3) WRITE BLOCK TO IPS FILE
C
0014          NR=1
0015          NSIZ=256
C
0016          IC = IURITU(NSIZ,BUFF(1,1),NR,ICHIPS)
0017          IF(IC.LT.0) IPUTFM = IC
```

```
C
0019        RETURN
0020        END
```

```
RT-11 FORTRAN IV        STORAGE MAP

NAME    OFFSET  ATTRIBUTES

IBUFF   000014  INTEGER*2 PARAMETER ARRAY (1)
IPUTFM  000016  INTEGER*2 VARIABLE
J       000022  INTEGER*2 VARIABLE
NR      000024  INTEGER*2 VARIABLE
NSIZ    000026  INTEGER*2 VARIABLE
IC      000030  INTEGER*2 VARIABLE
IWRITW  000000  INTEGER*2 PROCEDURE

COMMON BLOCK /IFILES/   LENGTH 001134

FSPEC   000000  INTEGER*2 ARRAY (39)
EXT     000116  INTEGER*2 ARRAY (2)
NRECS   000122  INTEGER*2 VARIABLE
NRECSZ  000124  INTEGER*2 VARIABLE
ICHIPS  000126  INTEGER*2 VARIABLE
IACT    000130  INTEGER*2 VARIABLE
MABREC  000132  INTEGER*2 VARIABLE
BUFF    000134  LOGICAL*1 ARRAY (512,1) VECTORED
INFL    000036  INTEGER*2 ARRAY (4)
OUTFL   000036  INTEGER*2 ARRAY (4)
*
```

```
C     ,                    FILE NAME: COML01.FOR
C?    ,
C     ,                    DATE LAST EDITED: 8-MAR-79
C
C
C     ,                    HISTORY:
C     ,
C     ,                    DATE:          MODIFICATION
C     ,                    -----          ------------
C     ,                    28-FEB-79      INITAL WRITING
C     ,                     8-MAR-79      CHANGE FUNCTION
C     ,                                   NAMES TO START WITH I
C
C     ,    COMTAL LIBRARY FUNCTIONS
C     ,    ------ ------- ---------
C
C     ,    THIS COMTAL LIBRARY CONSISTS OF A SERIES OF FUNCTION ROUTINES.
C     ,    THE FUNCTIONS PROVIDE BASIC BUILDING BLOCKS FOR USER PROGRAMS
C     ,    TO USE THE COMTAL.
C
C     ,    THE SOURCE FOR THES FUNCTIONS ARE IN 5 FILES. THEY ARE:
C     ,
C     ,    FILE NAME              FUNCTIONS
C     ,    ---------              ---------
C     ,    COML01.FOR             OPEN AND CLOSE
C     ,    COML02.FOR             TRANSFER CODE 1 TYPE
C     ,    COML03.FOR             TRANSFER CODE 2 TYPE
C     ,    COML04.FOR             TARGET POSITION
C     ,    COML05.FOR             TRANSFER CODE 3 TYPE
C
C     ,    THE FOLLOWING FUNCTIONS ARE PROVIDED:
C?    ,
C      NUMBER      NAME    FUNCTION
C     ,    ------  ----    --------
C
C     ,    N/A     CMOPEN  START OF COMTAL COMUNICATION
C     ,    N/A     CMCLOS  END   OF COMTAL COMUNICATION
C
C     ,     1      IWIM    WRITE  IMAGE    MEMORY
C     ,     2              NOT    DEFINED
C     ,     3      IRIM    READ   IMAGE    MEMORY
C     ,     4      IWGM    WRITE  GRAPHIC  MEMORY
C     ,     5              NOT    DEFINED
C     ,     6      IRIM    READ   GRAPHIC  MEMORY
C     ,     7      IWPS    WRITE  PSEUDO   MEMORY
```

```
C     .      8      IRPS    READ   PSEUDO   MEMORY
C     .      9      IWFM    WRITE  FUNCTION MEMORY
C     .     10      IRFM    READ   FUNCTION MEMORY
C     .     11      IWTP    WRITE  TARGET   POSITION
C     .     12      IRTP    READ   TARGET   POSITION
C     .     13      IWDC    WRITE  DISPLAY  COMMAND
C     .     14              NOT    DEFINED
C     .     15      IWCB    WRITE  COMMAND  BLOCK
C     .     16      IRCB    READ   COMMAND  BLOCK
C     .     17      IWCD    WRITE  CODE     DATA
```



```
C     .     18      IRCD    READ   CODE     DATA
C     .     19      IWDB    WRITE  DATA     BLOCK
C     .     20      IRDB    READ   DATA     BLOCK
C
C     .     ALL OF THESE ROUTINES ARE INTEGER FUNCTIONS.
C     .     THE VALUE OF THE FUNCTION IS DETERMINED BY THE
C     .     STATUS OF THE ROUTINE.
C     .     IN GENERAL:
C     .     FUNCTION VALUE          CONDITION
C     .     -------- -----          ---------
C     .              0             SUCCESS
C     .             -1             ERROR IN READ/WRITE OF COMTAL
C
C     ***************************************************
C     ***************************************************
C     ---------------------------------------------------
C     ***************************************************
C
C     .     START OF THE LIBRARY ROUTINES
C
C
0001        INTEGER FUNCTION ICMOPE
C     .     FUNCTIONAL DESCRIPTION:
C     .           THIS ROUTINE MUST BE THE FIRST CALL IN A USER
C     .           PROGRAM FOR COMTAL SERVICE.  IT ESTABLISHES
```

```
C     .          A CHANNEL - ICHIN - FOR I/O TO THE CONTAL.
C     .
C     .          AFTER THIS ROUTINE IS CALLED, OTHER CONTAL SERVICE
C     .          ROUTINES MAY BE CALLED.  AFTER ALL PROCESSING IS
C     .          FINISHED, A CALL TO CMCLOS IS MADE TO CLOSE THE CHANNEL.
C     .
C     .    ERRORS:
C     .
C     .          1,2,3   :      ERROR IN FETCHING HANDLER
C     .          -1,-2   :      ERROR IN OPENING CONTAL HANDLER
C
C     **************************************************
C
C     DATA DEFINITION
C
0002        INTEGER TCUIM, TCRIM, TCUGM, TCRGM, TCUPS, TCRPS
```



```
0003        INTEGER TCUFM, TCRFM, TCUTP, TCRTP, TCUDC, TCRDC
0004        INTEGER TCUCB, TCRCB, TCUCD, TCRCD, TCUDB, TCRDB
C
C
0005        INTEGER INFILE(4),DDB(8,20)
C
0006        INTEGER TCC
C
0007        INTEGER DB1(8),DB2(8),DB3(8),DB4(8),DB5(8)
0008        INTEGER DB6(8),DB7(8),DB8(8),DB9(8),DB10(8)
0009        INTEGER DB11(8),DB12(8),DB13(8),DB14(8),DB15(8)
0010        INTEGER DB16(8),DB17(8),DB18(8),DB19(8),DB20(8)
C
0011        LOGICAL*1 SPACE
C
C
0012        EQUIVALENCE (DB1 ,DDB(1,1 ))
0013        EQUIVALENCE (DB2 ,DDB(1,2 ))
0014        EQUIVALENCE (DB3 ,DDB(1,3 ))
0015        EQUIVALENCE (DB4 ,DDB(1,4 ))
0016        EQUIVALENCE (DB5 ,DDB(1,5 ))
0017        EQUIVALENCE (DB6 ,DDB(1,6 ))
0018        EQUIVALENCE (DB7 ,DDB(1,7 ))
0019        EQUIVALENCE (DB8 ,DDB(1,8 ))
0020        EQUIVALENCE (DB9 ,DDB(1,9 ))
0021        EQUIVALENCE (DB10,DDB(1,10))
0022        EQUIVALENCE (DB11,DDB(1,11))
0023        EQUIVALENCE (DB12,DDB(1,12))
0024        EQUIVALENCE (DB13,DDB(1,13))
```

```
0025          EQUIVALENCE (DB14,DDB(1,14))
0026          EQUIVALENCE (DB15,DDB(1,15))
0027          EQUIVALENCE (DB16,DDB(1,16))
0028          EQUIVALENCE (DB17,DDB(1,17))
0029          EQUIVALENCE (DB18,DDB(1,18))
0030          EQUIVALENCE (DB19,DDB(1,19))
0031          EQUIVALENCE (DB20,DDB(1,20))
      C
      C
0032          COMMON /DDB/ DDB
0033          COMMON /TC/ TCWIM, TCRIM, TCUGM, TCRGM, TCWPS, TCRPS,
     1                    TCWFM, TCRFM, TCWTP, TCRTP, TCWDC, TCRDC,
     2                  TCWCB, TCRCB, TCWCD, TCRCD, TCWDB, TCRDB
      C
0034          COMMON /CHAN/ ICHIN
      C
0035          COMMON /COMVAR/ MAXIM,MAXLN,MAXGM
      C
0036          COMMON /DATAA/ TCC
      C
      C
      C
      C     ,    THE FOLLOWING DEFINE THE NUMBER OF COMTAL IMAGES, NUMBER OF
      C     ,    LINES INAN IMAGE, AND THE NUMBER OF GRAGHICS,
      C     ,    NOTE: ALL NUMBERS ARE 1 LESS ACTUAL
      C     ,    (0 - 2, IS 3 MEMORIES)
      C
0037          DATA  MAXIM /2/, MAXLN / 255 /, MAXGM / 2 /
```



```
      C
      C
      C
      C     ,    THE FOLLOWING TABLE DEFINES THE DDBS FOR THE
      C     ,    VARIOUS FUNCTIONS DB1 IS USED FOR FUNCTION 1
      C     ,    WHICH IS WRITE IMAGE MEMORY (WIM) THIS IS
      C     ,    DETERMINED BY TCWIM=1.
      C
      C
0038          DATA DB1  / '101000 ,'      0 ,'0 ,'0 , -128,' 2403,' 3,' 0 /
0039          DATA DB2  / '101000,'       0 ,'0 ,'0 , - 64,' 1403,' 3,' 0 /
0040          DATA DB3  / '  1000,'  100000 ,'0 ,'0 , -128,' 2405,' 3,' 0 /
0041          DATA DB4  / '100000,'    1000 ,'0 ,'0 , - 16,' 2403,' 3,'400 /
0042          DATA DB5  / '100000,'    1000 ,'0 ,'0 , -  8,' 2403,' 3,'400 /
0043          DATA DB6  / '     0,'  101000 ,'0 ,'0 , - 16,' 2405,' 3,'400 /
0044          DATA DB7  / '140000,'   20000 ,'0 ,'0 , -256,' 3003,'23,'400 /
0045          DATA DB8  / ' 40000,'   20000 ,'0 ,'0 , -256,' 3005,' 3,'400 /
0046          DATA DB9  / '110000,'   60000 ,'0 ,'0 , -256,' 3003,'17,' 0 /
0047          DATA DB10 / ' 10000,'  160000 ,'0 ,'0 , -256,' 3005,'17,' 0 /
```

```
0048        DATA DB11 / :     0,:       0 ,:0 ,:0 , -  1,: 1403,:777,:1 /
0049        DATA DB12 / :     0,: 100000 ,:0 ,:0 , -  2,: 3405,:777,:0 /
C
0050        DATA DB13 / :     0,:       0 ,:0 ,:0 , -  1,:    3,:777,: 1 /
C
0051        DATA DB14 / :     0,:       0 ,:0 ,:0 , -  1,:  403,: 3,:400 /
0052        DATA DB15 / :150000,:   20000 ,:0 ,:0 , -256,: 3403,: 0,:   0 /
0053        DATA DB16 / : 50000,:  120000 ,:0 ,:0 , -256,: 3405,: 0,:   0 /
0054        DATA DB17 / :110000,:   60000 ,:0 ,:0 , -256,: 3403,: 0,:   0 /
0055        DATA DB18 / : 10000,:  160000 ,:0 ,:0 , -256,: 3405,: 0,:   0 /
0056        DATA DB19 / :140000,:   30000 ,:0 ,:0 , - 16,: 3403,: 0,:   0 /
0057        DATA DB20 / : 40000,:  130000 ,:0 ,:0 , - 16,: 3405,: 0,:   0 /
C
CC
C
0058        DATA INFILE /3RCMD,3R   ,3R   ,3R   /
C
0059        DATA SPACE /' '/
C
CCC
C     ;    THE FOLLOWING TABLE DEFINES WHICH FUNCTIONS ARE
C     ;    FUNCTION 1,2 3,ETC.
C     ;
C     ;    EG. FUNCTION 9 IS WFM - WRITE FUNCTION MEMORY
C
0060        DATA TCWIM / 1 /,  TCRIM / 3 /
0061        DATA TCWGM / 4 /,  TCRGM / 6 /
0062        DATA TCWPS / 7 /,  TCRPS / 8 /
0063        DATA TCWFM / 9 /,  TCRFM /10 /
0064        DATA TCWTP /11 /,  TCRTP /12 /
0065        DATA TCWDC /13 /,  TCRDC /14 /
0066        DATA TCWCB /15 /,  TCRCB /16 /
0067        DATA TCWCD /17 /,  TCRCD /18 /
0068        DATA TCWDB /19 /,  TCRDB /20 /
C
CC
```



```
C     END OF DATA DEFINITION
C
C*****************************************************
C*****************************************************
C
CCC
C     ;    1) GET CHANNEL
C     ;    2) FETCH HANDLER
C     ;    3) LOOK UP CONTAL HANDLER ON CHANNEL ICHIN
C
```

```
C
C     ,    SET FUNCTION = SUCCESS
C
0069        ICMOPE = 0
C
C
C
C     ,    1)
C
0070        ICHIN=IGETC()
C
C     ,    2)
C
0071        IC=IFETCH(INFILE)
0072        IF(IC.NE.0) GO TO 100
C
C     ,    3)
C
0074        IC=LOOKUP(ICHIN,INFILE)
C
0075  100   IF(IC.LT.0) ICMOPE=IC
0077        RETURN
0078        END
```



```
RT-11 FORTRAN IV        STORAGE MAP

NAME    OFFSET  ATTRIBUTES

INFILE  000014  INTEGER*2 ARRAY (4)
ICMOPE  000024  INTEGER*2 VARIABLE
SPACE   000026  LOGICAL*1 VARIABLE
IGETC   000000  INTEGER*2 PROCEDURE
IC      000100  INTEGER*2 VARIABLE
IFETCH  000000  INTEGER*2 PROCEDURE
LOOKUP  000000  INTEGER*2 PROCEDURE

COMMON BLOCK /DDB/      LENGTH 001160

DDB     000000  INTEGER*2 ARRAY (8,20) VECTORED
DB1     000000  INTEGER*2 ARRAY (8)
DB2     000020  INTEGER*2 ARRAY (8)
DB3     000040  INTEGER*2 ARRAY (8)
DB4     000060  INTEGER*2 ARRAY (8)
DB5     000100  INTEGER*2 ARRAY (8)
DB6     000120  INTEGER*2 ARRAY (8)
DB7     000140  INTEGER*2 ARRAY (8)
DB8     000160  INTEGER*2 ARRAY (8)
DB9     000200  INTEGER*2 ARRAY (8)
DB10    000220  INTEGER*2 ARRAY (8)
DB11    000240  INTEGER*2 ARRAY (8)
DB12    000260  INTEGER*2 ARRAY (8)
```

```
DB13    000300  INTEGER*2 ARRAY (8)
DB14    000320  INTEGER*2 ARRAY (8)
DB15    000340  INTEGER*2 ARRAY (8)
DB16    000360  INTEGER*2 ARRAY (8)
DB17    000400  INTEGER*2 ARRAY (8)
DB18    000420  INTEGER*2 ARRAY (8)
DB19    000440  INTEGER*2 ARRAY (8)
DB20    000460  INTEGER*2 ARRAY (8)

COMMON BLOCK /TC/       LENGTH 000044

TCWIM   000000  INTEGER*2 VARIABLE
TCRIM   000002  INTEGER*2 VARIABLE
TCWGM   000004  INTEGER*2 VARIABLE
TCRGM   000006  INTEGER*2 VARIABLE
TCWPS   000010  INTEGER*2 VARIABLE
TCRPS   000012  INTEGER*2 VARIABLE
TCWFM   000014  INTEGER*2 VARIABLE
TCRFM   000016  INTEGER*2 VARIABLE
TCWTP   000020  INTEGER*2 VARIABLE
TCRTP   000022  INTEGER*2 VARIABLE
TCWDC   000024  INTEGER*2 VARIABLE
TCRDC   000026  INTEGER*2 VARIABLE
TCWCB   000030  INTEGER*2 VARIABLE
TCRCB   000032  INTEGER*2 VARIABLE
TCWCD   000034  INTEGER*2 VARIABLE
TCRCD   000036  INTEGER*2 VARIABLE
TCWDB   000040  INTEGER*2 VARIABLE
TCRDB   000042  INTEGER*2 VARIABLE

COMMON BLOCK /CHAN/     LENGTH 000002

ICHIN   000000  INTEGER*2 VARIABLE
```



```
RT-11 FORTRAN IV        STORAGE MAP

NAME    OFFSET  ATTRIBUTES

COMMON BLOCK /COMVAR/   LENGTH 000006

MAXIM   000000  INTEGER*2 VARIABLE
MAXLN   000002  INTEGER*2 VARIABLE
MAXGM   000004  INTEGER*2 VARIABLE

COMMON BLOCK /DATAA/    LENGTH 000002

TCC     000000  INTEGER*2 VARIABLE
```

```
C
C
0001        INTEGER FUNCTION ICMCLO
C
C       ,    FUNCTIONAL DESCRIPTION:
C
C       ,            THIS ROUTINE IS CALLED AFTER ALL COMTAL
C       ,            PROCESSING IS FINISHED, A CALL TO ICMOPE MUST
C       ,            HAVE BEEN CALLED PRIOR TO THIS CALL.
C
C       ,    ERRORS:
C
C       ,            NONE.
C
C
C       ********************************************
C
C       DATA DEFINITION
C
C
0002        INTEGER TCUIM, TCRIM, TCUGM, TCRGM, TCUPS, TCRPS
0003        INTEGER TCUFM, TCRFM, TCUTP, TCRTP, TCUDC, TCRDC
0004        INTEGER TCUCB, TCRCB, TCUCD, TCRCD, TCUDB, TCRDB
C
0005        INTEGER DDB(8,20)
0006        INTEGER TCC
C
C
0007        COMMON /DDB/ DDB
C
0008        COMMON /TC/ TCUIM, TCRIM, TCUGM, TCRGM, TCUPS, TCRPS,
           *               TCUFM, TCRFM, TCUTP, TCRTP, TCUDC, TCRDC,
           *             TCUCB, TCRCB, TCUCD, TCRCD, TCUDB, TCRDB
C
0009        COMMON /CHAN/ ICHIN
C
0010        COMMON /COMVAR/ MAXIM,MAXLN,MAXGM
C
0011        COMMON /DATAA/ TCC
C
C       END OF DATA DEFINITION
C
C       NOTE: ALL CONSTANTS ARE DEFINED IN FUNCTION ICMOPE
C       ----- ------------                              ------
C
C       ********************************************************
C
0012        ICMCLO=0
C
```

```
0013          CALL CLOSEC(ICHIN)
0014          RETURN
0015          END
```

```
RT-11 FORTRAN IV        STORAGE MAP

NAME    OFFSET  ATTRIBUTES

ICMCLO  000014  INTEGER*2 VARIABLE
CLOSEC  000000  REAL*4    PROCEDURE

COMMON BLOCK /DDB/      LENGTH 000500

DDB     000000  INTEGER*2 ARRAY (8,20) VECTORED

COMMON BLOCK /TC/       LENGTH 000044

TCWIM   000000  INTEGER*2 VARIABLE
TCRIM   000002  INTEGER*2 VARIABLE
TCWGM   000004  INTEGER*2 VARIABLE
TCRGM   000006  INTEGER*2 VARIABLE
TCWPS   000010  INTEGER*2 VARIABLE
TCRPS   000012  INTEGER*2 VARIABLE
TCWFM   000014  INTEGER*2 VARIABLE
TCRFM   000016  INTEGER*2 VARIABLE
TCWTP   000020  INTEGER*2 VARIABLE
TCRTP   000022  INTEGER*2 VARIABLE
TCWDC   000024  INTEGER*2 VARIABLE
TCRDC   000026  INTEGER*2 VARIABLE
TCWCB   000030  INTEGER*2 VARIABLE
TCRCB   000032  INTEGER*2 VARIABLE
TCWCD   000034  INTEGER*2 VARIABLE
TCRCD   000036  INTEGER*2 VARIABLE
TCWDB   000040  INTEGER*2 VARIABLE
TCRDB   000042  INTEGER*2 VARIABLE

COMMON BLOCK /CHAN/     LENGTH 000002

ICHIN   000000  INTEGER*2 VARIABLE

COMMON BLOCK /COMVAR/   LENGTH 000006

MAXIM   000000  INTEGER*2 VARIABLE
MAXLN   000002  INTEGER*2 VARIABLE
MAXGM   000004  INTEGER*2 VARIABLE

COMMON BLOCK /DATAA/    LENGTH 000002

TCC     000000  INTEGER*2 VARIABLE
#
```

```
C     .                    FILE NAME: COML02.FOR
C     .                    DATE LAST EDITED: 8-MAR-79
C     .                    HISTORY:
C     .                    DATE:          MODIFICATION
C     .                    -----          ------------
C     .                    28-FEB-79      INITAL WRITING
C     .                     8-MAR-79      CHANGE FUNCTION
C     .                                   NAMES TO START WITH I
C
0001        INTEGER FUNCTION IWIM(IMNO,IMLN,BUFF)
C
C           FUNCTIONAL DESCRIPTION:
C
C           THIS FUNCTION IS USED TO WRITE ONE LINE OF
C           DATA FROM USER MEMORY TO THE COMTAL IMAGE SELECTED.
C
C           PARAMETERS:
C
C           IMNO  -  THE IMAGE NUMBER (1-3)
C           IMLN  -  THE LINE NUMBER (1-256) 1 AT TOP, 256 AT BOTTOM
C           BUFF  -  USER BUFFER ADDRESS
C
C           ERRORS:
C
C            0  -  SUCCESS
C           -1  -  WRITE ERROR
C
C
C
C           ***************************************************
C           DATA DEFINITION
C
0002        LOGICAL*1 BUFF(256)
C
C
0003        INTEGER TCWIM, TCRIM, TCWGM, TCRGM, TCWPS, TCRPS
0004        INTEGER TCWFM, TCRFM, TCWTP, TCRTP, TCWDC, TCRDC
0005        INTEGER TCWCB, TCRCB, TCWCD, TCRCD, TCWDB, TCRDB
```

```
      C
      C
0006        INTEGER DDB(8,20)
0007        INTEGER TCC
      C
      C
      C
0008        COMMON /DDB/ DDB
      C
0009        COMMON /TC/ TCWIM, TCRIM, TCWGM, TCRGM, TCWPS, TCRPS,
```



```
     *              TCWFM, TCRFM, TCWTP, TCRTP, TCWDC, TCRDC,
     *            TCWCB, TCRCB, TCWCD, TCRCD, TCWDB, TCRDB
      C
0010        COMMON /CHAN/ ICHIN
      C
0011        COMMON /COMVAR/ MAXIM,MAXLN,MAXGM
      C
0012        COMMON /DATAA/ TCC
      C
      C     END OF DATA DEFINITION
      C     NOTE: ALL CONSTANTS ARE DEFINED IN FUNCTION CMOPEN
      C     ----- -------------                    ------
      C     ***********************************************************
      C
      C     ,    SET  FUNCTION CODE
      C
0013        TCC=TCWIM
      C
      C     ,    CALL TRANSFER CODE 1 COMMON ROUTINE
      C
0014        IWIM=ITC1(IMNO,IMLN,BUFF)
      C
0015        RETURN
0016        END
```

```
NAME    OFFSET  ATTRIBUTES

BUFF    000020  LOGICAL*1 PARAMETER ARRAY (256)
IWIM    000022  INTEGER*2 VARIABLE
IWNO    000014  INTEGER*2 PARAMETER VARIABLE
IWLN    000016  INTEGER*2 PARAMETER VARIABLE
ITC1    000000  INTEGER*2 PROCEDURE

COMMON BLOCK /DDB/      LENGTH 000500

DDB     000000  INTEGER*2 ARRAY (8,20) VECTORED

COMMON BLOCK /TC/       LENGTH 000044

TCUIM   000000  INTEGER*2 VARIABLE
TCRIM   000002  INTEGER*2 VARIABLE
TCUGM   000004  INTEGER*2 VARIABLE
TCRGM   000006  INTEGER*2 VARIABLE
TCUPS   000010  INTEGER*2 VARIABLE
TCRPS   000012  INTEGER*2 VARIABLE
TCUFM   000014  INTEGER*2 VARIABLE
TCRFM   000016  INTEGER*2 VARIABLE
TCUTP   000020  INTEGER*2 VARIABLE
TCRTP   000022  INTEGER*2 VARIABLE
TCUDC   000024  INTEGER*2 VARIABLE
TCRDC   000026  INTEGER*2 VARIABLE
TCUCB   000030  INTEGER*2 VARIABLE
TCRCB   000032  INTEGER*2 VARIABLE
TCUCD   000034  INTEGER*2 VARIABLE
TCRCD   000036  INTEGER*2 VARIABLE
TCUDB   000040  INTEGER*2 VARIABLE
TCRDB   000042  INTEGER*2 VARIABLE

COMMON BLOCK /CHAN/     LENGTH 000002

ICHIN   000000  INTEGER*2 VARIABLE

COMMON BLOCK /COMVAR/   LENGTH 000006

MAXIM   000000  INTEGER*2 VARIABLE
MAXLN   000002  INTEGER*2 VARIABLE
MAXGM   000004  INTEGER*2 VARIABLE

COMMON BLOCK /DATAA/    LENGTH 000002

TCC     000000  INTEGER*2 VARIABLE



RT-11 FORTRAN IV        V01B-08A   FRI 09-MAR-79 16:16:03      PAGE 001
      C
```

```
0001        INTEGER FUNCTION IRIM(IMNO,IMLN,BUFF)
C
C     FUNCTIONAL DESCRIPTION:
C
C             THIS FUNCTION IS USED TO READ ONE LINE OF
C             DATA FROM THE COMTAL IMAGE SELECTED TO THE
C             USER BUFFER.
C
C     PARAMETERS:
C
C             IMNO  =  THE IMAGE NUMBER (1-3)
C             IMLN  =  THE LINE NUMBER (1-256) 1 AT TOP, 256 AT BOTTOM
C             BUFF  =  USER BUFFER ADDRESS
C
C     ERRORS:
C
C              0  =  SUCCESS
C             -1  =  WRITE ERROR
C
C
C
C**********************************************
C
C     DATA DEFINITION
C
0002        LOGICAL*1 BUFF(256)
C
C
0003        INTEGER TCWIM, TCRIM, TCWGM, TCRGM, TCWPS, TCRPS
0004        INTEGER TCWFM, TCRFM, TCWTP, TCRTP, TCWDC, TCRDC
0005        INTEGER TCWCB, TCRCB, TCWCD, TCRCD, TCWDB, TCRDB
C
0006        INTEGER DDB(8,20)
0007        INTEGER TCC
C
C
0008        COMMON /DDB/ DDB
C
0009        COMMON /TC/ TCWIM, TCRIM, TCWGM, TCRGM, TCWPS, TCRPS,
     *                  TCWFM, TCRFM, TCWTP, TCRTP, TCWDC, TCRDC,
     *                TCWCB, TCRCB, TCWCD, TCRCD, TCWDB, TCRDB
C
0010        COMMON /CHAN/ ICHIN
C
0011        COMMON /COMVAR/ MAXIM,MAXLN,MAXGM
C
0012        COMMON /DATAA/ TCC
C
C     END OF DATA DEFINITION
C
C     NOTE: ALL CONSTANTS ARE DEFINED IN FUNCTION CMOPEN
```

```
RT-11 FORTRAN IV        V01B-08A   FRI 09-MAR-79 16:16:03       PAGE 002
      C     ----- -------------                          ------
      C
      C     ***************************************************
      C
      C
      C     ,    SET  FUNCTION CODE
      C
      C
0013          TCC=TCRIM
      C
      C
      C     ,    CALL TRANSFER CODE 1 COMMON ROUTINE
      C
0014          IRIM=ITC1(IMNO,IMLN,BUFF)
      C
0015          RETURN
0016          END
```



```
RT-11 FORTRAN IV        STORAGE MAP

NAME    OFFSET  ATTRIBUTES

BUFF    000020  LOGICAL*1 PARAMETER ARRAY (256)
IRIM    000022  INTEGER*2 VARIABLE
IMNO    000014  INTEGER*2 PARAMETER VARIABLE
IMLN    000016  INTEGER*2 PARAMETER VARIABLE
ITC1    000000  INTEGER*2 PROCEDURE

COMMON BLOCK /DDB/      LENGTH 000500

DDB     000000  INTEGER*2 ARRAY (8,20) VECTORED

COMMON BLOCK /TC/       LENGTH 000044

TCWIM   000000  INTEGER*2 VARIABLE
TCRIM   000002  INTEGER*2 VARIABLE
TCWGM   000004  INTEGER*2 VARIABLE
TCRGM   000006  INTEGER*2 VARIABLE
TCWPS   000010  INTEGER*2 VARIABLE
TCPPS   000012  INTEGER*2 VARIABLE
TCWFM   000014  INTEGER*2 VARIABLE
TCRFM   000016  INTEGER*2 VARIABLE
TCUTP   000020  INTEGER*2 VARIABLE
```

```
TCRTP   000022  INTEGER*2 VARIABLE
TCLCC   000024  INTEGER*2 VARIABLE
TCRDC   000026  INTEGER*2 VARIABLE
TCLCB   000030  INTEGER*2 VARIABLE
TCRCB   000032  INTEGER*2 VARIABLE
TCLCD   000034  INTEGER*2 VARIABLE
TCRCD   000036  INTEGER*2 VARIABLE
TCLDB   000040  INTEGER*2 VARIABLE
TCRDB   000042  INTEGER*2 VARIABLE

COMMON BLOCK /CHAN/     LENGTH 000002

ICHIN   000000  INTEGER*2 VARIABLE

COMMON BLOCK /COMVAR/   LENGTH 000006

MAXIM   000000  INTEGER*2 VARIABLE
MAXLM   000002  INTEGER*2 VARIABLE
MAXGM   000004  INTEGER*2 VARIABLE

COMMON BLOCK /DATAA/    LENGTH 000002

TCC     000000  INTEGER*2 VARIABLE
```



```
0001      C
          C
          C     INTEGER FUNCTION IWGM(IMNO,IMLN,BUFF)
          C
          C
          C     ,    FUNCTIONAL DESCRIPTION:
          C     ,
          C     ,         THIS FUNCTION IS USED TO WRITE ONE LINE OF
          C     ,         DATA FROM USER MEMORY TO THE COMTAL GRAPHIC SELECTED.
          C     ,
          C     ,    PARAMETERS:
          C     ,
          C     ,         IMNO  -  THE GRAPHIC NUMBER (1-3)
          C     ,         IMLN  -  THE LINE NUMBER (1-256) 1 AT TOP, 256 AT BOTTOM
          C     ,         BUFF  -  USER BUFFER ADDRESS
          C     ,
          C     ,    ERRORS:
          C     ,
          C     ,          0  :  SUCCESS
          C     ,         -1  :  WRITE ERROR
          C
          C
          C
          C     ***************************************************
```

```
C
C     DATA DEFINITION
C
0002        LOGICAL*1 BUFF(256)
C
C
0003        INTEGER TCUIM, TCRIM, TCUGM, TCRGM, TCUPS, TCRPS
0004        INTEGER TCUFM, TCRFM, TCUTP, TCRTP, TCUDC, TCRDC
0005        INTEGER TCUCB, TCRCB, TCUCD, TCRCD, TCUDB, TCRDB
C
C
0006        INTEGER DDB(8,20)
0007        INTEGER TCC
C
C
C
0008        COMMON /DDB/ DDB
C
0009        COMMON /TC/ TCUIM, TCRIM, TCUGM, TCRGM, TCUPS, TCRPS,
           *               TCUFM, TCRFM, TCUTP, TCRTP, TCUDC, TCRDC,
           *             TCUCB, TCRCB, TCUCD, TCRCD, TCUDB, TCRDB
C
0010        COMMON /CHAN/ ICHIN
C
0011        COMMON /COMVAR/ MAXIM,MAXLN,MAXGM
C
0012        COMMON /DATAA/ TCC
C
C     END OF DATA DEFINITION
C
C     NOTE: ALL CONSTANTS ARE DEFINED IN FUNCTION CMOPEN
C     ----- -------------                 ------
```



```
C
C     ***********************************************************
C
C
C     ,    SET  FUNCTION CODE
C
C
0013        TCC=TCUGM
C
C
C     ,    CALL TRANSFER CODE 1 COMMON ROUTINE
C
0014        IUGM=ITC1(IMNO,IMLN,BUFF)
C
0015        RETURN
0016        END
```

```
RT-11 FORTRAN IV        STORAGE MAP

NAME    OFFSET  ATTRIBUTES

BUFF    000020  LOGICAL*1 PARAMETER ARRAY (256)
IWGM    000022  INTEGER*2 VARIABLE
IMNO    000014  INTEGER*2 PARAMETER VARIABLE
IMLN    000016  INTEGER*2 PARAMETER VARIABLE
ITC1    000000  INTEGER*2 PROCEDURE

COMMON BLOCK /DDB/      LENGTH 000500

DDB     000000  INTEGER*2 ARRAY (8,20) VECTORED

COMMON BLOCK /TC/       LENGTH 000044

TCWIM   000000  INTEGER*2 VARIABLE
TCPIM   000002  INTEGER*2 VARIABLE
TCWGM   000004  INTEGER*2 VARIABLE
TCRGM   000006  INTEGER*2 VARIABLE
TCWPS   000010  INTEGER*2 VARIABLE
TCRPS   000012  INTEGER*2 VARIABLE
TCWFM   000014  INTEGER*2 VARIABLE
TCRFM   000016  INTEGER*2 VARIABLE
TCWTP   000020  INTEGER*2 VARIABLE
TCRTP   000022  INTEGER*2 VARIABLE
TCWDC   000024  INTEGER*2 VARIABLE
TCRDC   000026  INTEGER*2 VARIABLE
TCWCS   000030  INTEGER*2 VARIABLE
TCRCB   000032  INTEGER*2 VARIABLE
TCWCD   000034  INTEGER*2 VARIABLE
TCRCD   000036  INTEGER*2 VARIABLE
TCWDB   000040  INTEGER*2 VARIABLE
TCRDB   000042  INTEGER*2 VARIABLE

COMMON BLOCK /CHAN/     LENGTH 000002

ICHIN   000000  INTEGER*2 VARIABLE

COMMON BLOCK /COMVAR/   LENGTH 000006

MW/IM   000000  INTEGER*2 VARIABLE
MW/LN   000002  INTEGER*2 VARIABLE
MW/GM   000004  INTEGER*2 VARIABLE

COMMON BLOCK /DATAA/    LENGTH 000002

TCC     000000  INTEGER*2 VARIABLE
```

```
0001      C
          INTEGER FUNCTION IRGM(IMNO,IMLN,BUFF)
          C
          C
          C     FUNCTIONAL DESCRIPTION:
          C
          C             THIS FUNCTION IS USED TO READ ONE LINE OF
          C             DATA FROM THE COMTAL GRAPHIC SELECTED TO
          C             THE USER BUFFER.
          C
          C     PARAMETERS:
          C
          C             IMNO  =  THE GRAPHIC NUMBER (1-3)
          C             IMLN  =  THE LINE NUMBER (1-256) 1 AT TOP, 256 AT BOTTOM
          C             BUFF  =  USER BUFFER ADDRESS
          C
          C     ERRORS:
          C
          C              0  =  SUCCESS
          C             -1  =  WRITE ERROR
          C
          C
          C
          C
          C**********************************************
          C     DATA DEFINITION
          C
0002      C     LOGICAL*1 BUFF(256)
          C
          C
0003      C     INTEGER TCWIM, TCRIM, TCWGM, TCRGM, TCWPS, TCRPS
0004      C     INTEGER TCWFM, TCRFM, TCWTP, TCRTP, TCWDC, TCRDC
0005      C     INTEGER TCWCB, TCRCB, TCWCD, TCRCD, TCWDB, TCRDB
          C
          C
0006      C     INTEGER DDB(8,20)
0007      C     INTEGER TCC
          C
          C
          C
0008      C     COMMON /DDB/ DDB
          C
0009      C     COMMON /TC/ TCWIM, TCRIM, TCWGM, TCRGM, TCWPS, TCRPS,
          C    *            TCWFM, TCRFM, TCWTP, TCRTP, TCWDC, TCRDC,
          C    *            TCWCB, TCRCB, TCWCD, TCRCD, TCWDB, TCRDB
          C
0010      C     COMMON /CHAN/ ICHIN
          C
```

```
0011          COMMON /COMVAR/ MAXIM,MAXLN,MAXGM
0012      C   COMMON /DATAA/ TCC
          C
          C   END OF DATA DEFINITION
          C
          C   NOTE: ALL CONSTANTS ARE DEFINED IN FUNCTION CMOPEN
```

```
RT-11 FORTRAN IV        V01B-08A   FRI 09-MAR-79 16:16:07       PAGE 002
          C     ----- -------------                          ------
          C
          C     **********************************************************
          C
          C     ,    SET  FUNCTION CODE
          C
0013          TCC=TCRGM
          C
          C     ,    CALL TRANSFER CODE 1 COMMON ROUTINE
          C
0014          IRGM=ITC1(IMNO,IMLN,BUFF)
          C
0015          RETURN
0016          END
```

```
RT-11 FORTRAN IV        STORAGE MAP

NAME    OFFSET  ATTRIBUTES

BUFF    000020  LOGICAL*1 PARAMETER ARRAY (256)
IRGM    000022  INTEGER*2 VARIABLE
IMNO    000014  INTEGER*2 PARAMETER VARIABLE
IMLN    000016  INTEGER*2 PARAMETER VARIABLE
ITC1    000000  INTEGER*2 PROCEDURE

COMMON BLOCK /DDB/      LENGTH 000500

DDB     000000  INTEGER*2 ARRAY (8,20) VECTORED

COMMON BLOCK /TC/       LENGTH 000044
```

```
TCUIN   000000  INTEGER*2 VARIABLE
TCRIM   000002  INTEGER*2 VARIABLE
TCWGN   000004  INTEGER*2 VARIABLE
TCRGM   000006  INTEGER*2 VARIABLE
TCWPS   000010  INTEGER*2 VARIABLE
TCRPS   000012  INTEGER*2 VARIABLE
TCWFN   000014  INTEGER*2 VARIABLE
TCRFN   000016  INTEGER*2 VARIABLE
TCWTP   000020  INTEGER*2 VARIABLE
TCRTP   000022  INTEGER*2 VARIABLE
TCWDC   000024  INTEGER*2 VARIABLE
TCRDC   000026  INTEGER*2 VARIABLE
TCWCB   000030  INTEGER*2 VARIABLE
TCRCB   000032  INTEGER*2 VARIABLE
TCWCD   000034  INTEGER*2 VARIABLE
TCRCD   000036  INTEGER*2 VARIABLE
TCWDB   000040  INTEGER*2 VARIABLE
TCRDB   000042  INTEGER*2 VARIABLE

COMMON BLOCK /CHAN/     LENGTH 000002

ICHIN   000000  INTEGER*2 VARIABLE

COMMON BLOCK /COMVAR/   LENGTH 000006

MAXIM   000000  INTEGER*2 VARIABLE
MAXLN   000002  INTEGER*2 VARIABLE
MAXGM   000004  INTEGER*2 VARIABLE

COMMON BLOCK /DATAA/    LENGTH 000002

TCC     000000  INTEGER*2 VARIABLE
```



```
RT-11 FORTRAN IV        V01B-08A   FRI 09-MAR-79 16:16:09       PAGE 001

      C
0001        INTEGER FUNCTION ITC1(IMNO,IMLN,BUFF)
      C
      C
      C     ,    FUNCTIONAL DESCRIPTION:
      C
      C     ,
      C     ,           THIS FUNCTION IS CALLED BY ALL TRANSFER CODE 1
      C     ,           TYPE COMMANDS (READ/WRITE IMAGE AND GRAPHIC
      C     ,           MEMORIES)
      C
      C     ,
      C     ,           THE CALLING ROUTINE SETS THE FUNCTION NUMBER
      C     ,           IN TCC.  THE ROUTINE CORRECTS THE LINE NUMBER,
      C     ,           AND IMAGE/GRAPHIC NUMBER,  FORCES THE LINE
      C     ,           NUMBER < 256, IMAGE NO. <3, AND THEN CALLES
      C     ,           THE HANDLER.
      C
```

```
C
C     .    PARAMETERS:
C     .
C     .            IMNO  =  THE IMAGE/GRAPHIC NUMBER (1-3)
C     .            IMLN  =  THE LINE NUMBER (1-256) 1 AT TOP,
C     .                     256 AT BOTTOM
C     .            BUFF  =  USER BUFFER ADDRESS
C     .
C     .    ERRORS:
C     .
C     .             0  =  SUCCESS
C     .            -1  =  WRITE ERROR
C
C
C
C     ***********************************************
C     DATA DEFINITION
C
0002        LOGICAL*1 BUFF(256)
C
C
0003        INTEGER TCWIM, TCRIM, TCWGM, TCRGM, TCWPS, TCRPS
0004        INTEGER TCWFM, TCRFM, TCWTP, TCRTP, TCWDC, TCRDC
0005        INTEGER TCWCB, TCRCB, TCWCD, TCRCD, TCWDB, TCRDB
C
C
0006        INTEGER DDB(8,20)
0007        INTEGER TCC
C
C
0008        COMMON /DDB/ DDB
C
0009        COMMON /TC/ TCWIM, TCRIM, TCWGM, TCRGM, TCWPS, TCRPS,
           *              TCWFM, TCRFM, TCWTP, TCRTP, TCWDC, TCRDC,
           *            TCWCB, TCRCB, TCWCD, TCRCD, TCWDB, TCRDB
C
0010        COMMON /CHAN/ ICHIN


PT-11 FORTRAN IV        V01B-08A   FRI 09-MAR-79 16:16:09       PAGE 002

C
0011        COMMON /COMVAR/ MAXIM,MAXLN,MAXGM
C
0012        COMMON /DATAA/ TCC
C
C
C           END OF DATA DEFINITION
C
```

```
C     NOTE: ALL CONSTANTS ARE DEFINED IN FUNCTION CMOPEN
C     ----- -------------                    ------
C
C     ***********************************************************
C
C
C     CORRECT IMAGE NUMBER, CORRECT LINE NUMBER
C
0013        NO=IMNO-1
0014        LN=IMLN-1
C
C
C     FORCE IMAGE/GRAPHIC NUMBER, 0 <= IMAGE/GRAPHIC NO. < 2.
C     FORCE LINE NUMBER,          0 <= LINE NO.          < 256
C
0015        IF((NO.GT.MAXIM).OR.(NO.LT.0)) NO=0
0017        IF((LN.GT.MAXLN).OR.(LN.LT.0)) LN=0
C
C     SET ITC1=SUCCESS
C
0019        ITC1=0
C
C     GET USER BUFFER ADDRESS IN DDB
C
0020        DDB(4,TCC)=IADDR(BUFF)
C
C     SET TRANSFER COMMAND: SET IMAGE NUMBER, ADD LINE NUMBER
C
0021        DDB(3,TCC)=NO%*4000
0022        DDB(3,TCC)=DDB(3,TCC)+LN
C
C     CALL HANDLER
C
0023        IC=IWRITW(256,DDB(1,TCC),1,ICHIN)
C
C     CHECK FOR ERROR
C
0024        IF(IC.LT.0) ITC1=IC
0026        RETURN
0027        END
```

```
RT-11 FORTRAN IV        STORAGE MAP

NAME    OFFSET  ATTRIBUTES

BUFF    000020  LOGICAL*1 PARAMETER ARRAY (256)
ITC1    000022  INTEGER*2 VARIABLE
IMNO    000014  INTEGER*2 PARAMETER VARIABLE
IMLN    000016  INTEGER*2 PARAMETER VARIABLE
NO      000100  INTEGER*2 VARIABLE
LN      000102  INTEGER*2 VARIABLE
```

```
IADDR   000000  INTEGER*2 PROCEDURE
IC      000104  INTEGER*2 VARIABLE
IUPITU  000000  INTEGER*2 PROCEDURE

COMMON BLOCK /DDB/      LENGTH 000500

DDB     000000  INTEGER*2 ARRAY (8,20) VECTORED

COMMON BLOCK /TC/       LENGTH 000044

TCUIM   000000  INTEGER*2 VARIABLE
TCRIM   000002  INTEGER*2 VARIABLE
TCUGM   000004  INTEGER*2 VARIABLE
TCRGM   000006  INTEGER*2 VARIABLE
TCUPS   000010  INTEGER*2 VARIABLE
TCRPS   000012  INTEGER*2 VARIABLE
TCUFM   000014  INTEGER*2 VARIABLE
TCRFM   000016  INTEGER*2 VARIABLE
TCUTP   000020  INTEGER*2 VARIABLE
TCRTP   000022  INTEGER*2 VARIABLE
TCUDC   000024  INTEGER*2 VARIABLE
TCRDC   000026  INTEGER*2 VARIABLE
TCUCB   000030  INTEGER*2 VARIABLE
TCRCB   000032  INTEGER*2 VARIABLE
TCUCD   000034  INTEGER*2 VARIABLE
TCRCD   000036  INTEGER*2 VARIABLE
TCUDB   000040  INTEGER*2 VARIABLE
TCRDB   000042  INTEGER*2 VARIABLE

COMMON BLOCK /CHAN/     LENGTH 000002

ICHIN   000000  INTEGER*2 VARIABLE

COMMON BLOCK /COMVAR/   LENGTH 000006

MAXIM   000000  INTEGER*2 VARIABLE
MAXLN   000002  INTEGER*2 VARIABLE
MAXGM   000004  INTEGER*2 VARIABLE

COMMON BLOCK /DATAA/    LENGTH 000002

TCC     000000  INTEGER*2 VARIABLE
```

```
C     .                    FILE NAME: COML03.FOR
C     .
C     .                    DATE LAST EDITED: 21-MAR-79
C     .
C     .                    HISTORY:
C     .
C     .                    DATE:          MODIFICATION
C     .                    -----          ------------
C     .                    28-FEB-79      INITAL WRITING
C     .                    19-MAR-79      CHANGE FUNCTIONS
C     .                                   TO START WITH I
C     .                    21-MAR-79      CHANGE ALL 'FMNO'
C     .                                   TO 'IFMNO'
C
0001        INTEGER FUNCTION IWFM(IFMNO,BUFF)
C
C           FUNCTIONAL DESCRIPTION:
C
C           THIS ROUTINE WRITES THE SELECTED FUNCTION
C           MEMORY FROM THE USER BUFFER.
C
C           PARAMETERS:
C
C           IFMNO  =  FUNCTION MEMORY NUMBER (1-3)
C           BUFF   =  USER BUFFER (INTEGER ARRAY 256)
C
C           ERRORS:
C
C            0 =  SUCCESS
C           -1 =  WRITE  ERROR
C
C
C           ***********************************************
C
C           DATA DEFINITION
C
0002        INTEGER     BUFF(256)
C
C
0003        INTEGER TCWIM, TCRIM, TCWGM, TCRGM, TCWPS, TCRPS
0004        INTEGER TCWFM, TCRFM, TCWTP, TCRTP, TCWDC, TCRDC
0005        INTEGER TCWCB, TCRCB, TCWCD, TCRCD, TCWDB, TCRDB
C
C
0006        INTEGER DDB(8,20)
```

```
0007          INTEGER TCC
              C
              C
0008          COMMON /DDB/ DDB
              C
0009          COMMON /TC/ TCUIN, TCRIN, TCUGM, TCRGM, TCUPS, TCRPS,
             1                TCUFM, TCRFM, TCUTP, TCRTP, TCUDC, TCRDC,
             2             TCUCB, TCRCB, TCUCD, TCRCD, TCUDB, TCRDB
              C
```

RT-11 FORTRAN IV V01B-08A WED 21-MAR-79 16:28:39 PAGE 002

```
0010          COMMON /CHAN/ ICHIN
              C
0011          COMMON /COMVAR/ MAXIM,MAXLN,MAXGM
              C
0012          COMMON /DATAA/ TCC
              C
              C     END OF DATA DEFINITION
              C
              C     NOTE: ALL CONSTANTS ARE DEFINED IN FUNCTION CMOPEN
              C     ----- ------------                  ------
              C
              C     ***************************************************
              C
              C
              C     1) SET FUNCTION NUMBER
              C     2) CALL TRANSFER CODE 2 COMMON ROUTINE
              C
0013          TCC=TCUFM
              C
0014          IUFM=ITC2(IFMNO,BUFF)
              C
0015          RETURN
0016          END
```

RT-11 FORTRAN IV STORAGE MAP

NAME	OFFSET	ATTRIBUTES
BUFF	000016	INTEGER*2 PARAMETER ARRAY (256)
IUFM	000020	INTEGER*2 VARIABLE
IFMNO	000014	INTEGER*2 PARAMETER VARIABLE

```
ITS2    000000  INTEGER*2 PROCEDURE

COMMON BLOCK /DDB/      LENGTH 000500

DDB     000000  INTEGER*2 ARRAY (8,20) VECTORED

COMMON BLOCK /TC/       LENGTH 000044

TCUIM   000000  INTEGER*2 VARIABLE
TCRIM   000002  INTEGER*2 VARIABLE
TCUGM   000004  INTEGER*2 VARIABLE
TCRGM   000006  INTEGER*2 VARIABLE
TCUPS   000010  INTEGER*2 VARIABLE
TCPPS   000012  INTEGER*2 VARIABLE
TCUFM   000014  INTEGER*2 VARIABLE
TCPFM   000016  INTEGER*2 VARIABLE
TCUTP   000020  INTEGER*2 VARIABLE
TCRTP   000022  INTEGER*2 VARIABLE
TCUDC   000024  INTEGER*2 VARIABLE
TCRDC   000026  INTEGER*2 VARIABLE
TCUCB   000030  INTEGER*2 VARIABLE
TCRCB   000032  INTEGER*2 VARIABLE
TCUCD   000034  INTEGER*2 VARIABLE
TCPCD   000036  INTEGER*2 VARIABLE
TCUDB   000040  INTEGER*2 VARIABLE
TCRDB   000042  INTEGER*2 VARIABLE

COMMON BLOCK /CHAN/     LENGTH 000002

ICHIN   000000  INTEGER*2 VARIABLE

COMMON BLOCK /COMVAR/   LENGTH 000006

MAXIM   000000  INTEGER*2 VARIABLE
MAXLN   000002  INTEGER*2 VARIABLE
MAXGM   000004  INTEGER*2 VARIABLE

COMMON BLOCK /DATAA/    LENGTH 000002

TCC     000000  INTEGER*2 VARIABLE
```



```
PT-11 FORTRAN IV        V01B-08A   WED 21-MAR-79 16:28:41       PAGE 001

                C
0001            INTEGER FUNCTION IRFM(IFMNO,BUFF)
                C
                C       FUNCTIONAL DESCRIPTION:
                C
                C       THIS ROUTINE READS THE SELECTED FUNCTION
                C       MEMORY AND FILLS THE USER BUFFER.
                C
```

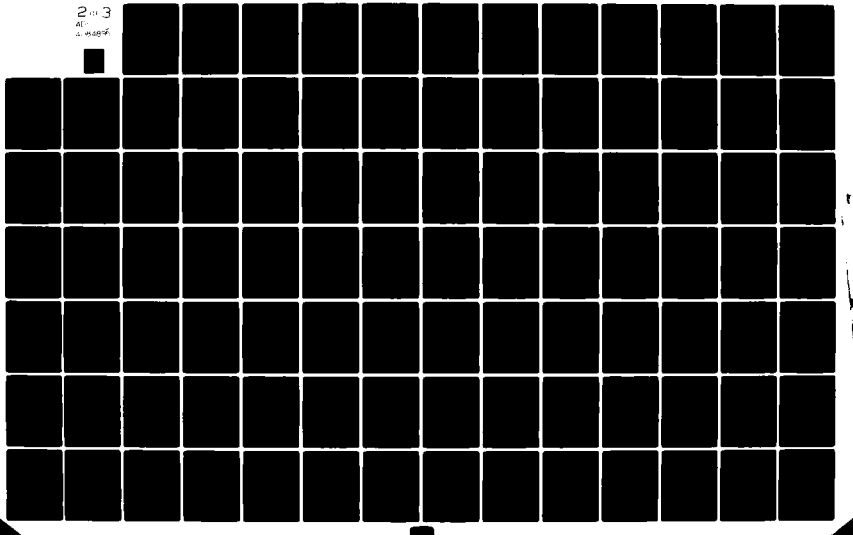
```
C
C     PARAMETERS:
C
C     IFMNO  -  FUNCTION MEMORY NUMBER (1-3)
C     BUFF   -  USER BUFFER (INTEGER ARRAY 256)
C
C     ERRORS:
C
C      0 -  SUCCESS
C     -1 -  WRITE  ERROR
C
C
C*****************************************************
C
C     DATA DEFINITION
C
0002  C     INTEGER     BUFF(256)
C
C
0003  C     INTEGER TCUIM, TCRIM, TCUGM, TCRGM, TCUPS, TCRPS
0004  C     INTEGER TCUFM, TCRFM, TCUTP, TCRTP, TCUDC, TCRDC
0005  C     INTEGER TCUCB, TCRCB, TCUCD, TCRCD, TCUDB, TCRDB
C
0006  C     INTEGER DDB(8,20)
0007  C     INTEGER TCC
C
C
0008  C     COMMON /DDB/ DDB
C
0009  C     COMMON /TC/ TCUIM, TCRIM, TCUGM, TCRGM, TCUPS, TCRPS,
     *                  TCUFM, TCRFM, TCUTP, TCRTP, TCUDC, TCRDC,
     *                  TCUCB, TCRCB, TCUCD, TCRCD, TCUDB, TCRDB
C
0010  C     COMMON /CHAN/ ICHIN
C
0011  C     COMMON /COMVAR/ MAXIM,MAXLN,MAXGM
C
0012  C     COMMON /DATAA/ TCC
C
C     END OF DATA DEFINITION
C
C     NOTE: ALL CONSTANTS ARE DEFINED IN FUNCTION CMOPEN
C     ----- -------------                            ------
C
C*****************************************************
C
```

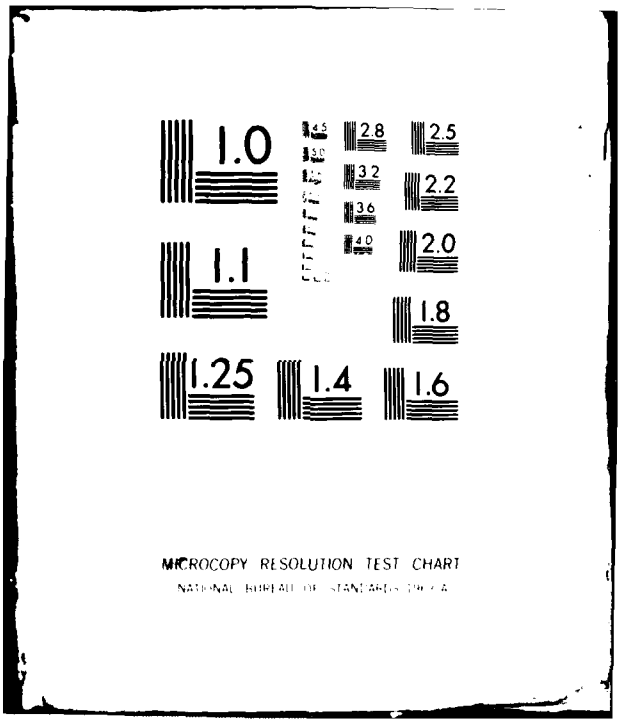
```
C     1) SET FUNCTION NUMBER
C     2) CALL TRANSFER CODE 2 COMMON ROUTINE
C
0013        TCC=TCRFM
C
0014        IRFM=ITC2(IFMNO,BUFF)
C
0015        RETURN
0016        END
```

```
RT-11 FORTRAN IV        STORAGE MAP

NAME    OFFSET  ATTRIBUTES

BUFF    000016  INTEGER*2 PARAMETER ARRAY (256)
IRFM    000020  INTEGER*2 VARIABLE
IFMNO   000014  INTEGER*2 PARAMETER VARIABLE
ITC2    000000  INTEGER*2 PROCEDURE

COMMON BLOCK /DDB/      LENGTH 000500

DDB     000000  INTEGER*2 ARRAY (8,20) VECTORED

COMMON BLOCK /TC/       LENGTH 000044

TCUIM   000000  INTEGER*2 VARIABLE
TCPIM   000002  INTEGER*2 VARIABLE
TCWGM   000004  INTEGER*2 VARIABLE
TCRGM   000006  INTEGER*2 VARIABLE
TCWPS   000010  INTEGER*2 VARIABLE
TCRPS   000012  INTEGER*2 VARIABLE
TCWFM   000014  INTEGER*2 VARIABLE
TCPFM   000016  INTEGER*2 VARIABLE
TCWTP   000020  INTEGER*2 VARIABLE
TCRTP   000022  INTEGER*2 VARIABLE
TCWDC   000024  INTEGER*2 VARIABLE
TCRDC   000026  INTEGER*2 VARIABLE
TCWCB   000030  INTEGER*2 VARIABLE
TCRCB   000032  INTEGER*2 VARIABLE
TCWCD   000034  INTEGER*2 VARIABLE
TCRCD   000036  INTEGER*2 VARIABLE
TCWDB   000040  INTEGER*2 VARIABLE
TCRDB   000042  INTEGER*2 VARIABLE

COMMON BLOCK /CHAN/     LENGTH 000002

ICHIN   000000  INTEGER*2 VARIABLE

COMMON BLOCK /COMVAR/   LENGTH 000006
```

```
MAIN   000000  INTEGER*2 VARIABLE
MAIN   000002  INTEGER*2 VARIABLE
MAIN   000004  INTEGER*2 VARIABLE

COMMON BLOCK /DATAA/    LENGTH 000002

TCC    000000  INTEGER*2 VARIABLE



RT-11 FORTRAN IV        V01B-08A   WED 21-MAR-79 16:28:44       PAGE 001

      C
0001        INTEGER FUNCTION IWPS(BUFF)
      C
      C     FUNCTIONAL DESCRIPTION:
      C
      C     THIS ROUTINE WRITES THE PSEUDO-COLOR
      C     MEMORY FROM THE USER BUFFER.
      C
      C     PARAMETERS:
      C
      C     BUFF  -  USER BUFFER (INTEGER ARRAY 256)
      C
      C     ERRORS:
      C
      C     0 -  SUCCESS
      C    -1 -  WRITE  ERROR
      C
      C
      C    ***********************************************
      C
      C     DATA DEFINITION
      C
0002        INTEGER     BUFF(256)
      C
      C
0003        INTEGER TCUIM, TCRIM, TCUGM, TCRGM, TCUPS, TCRPS
0004        INTEGER TCUFM, TCRFM, TCUTP, TCRTP, TCUDC, TCRDC
0005        INTEGER TCUCB, TCRCB, TCUCD, TCRCD, TCUDB, TCRDB
      C
      C
0006        INTEGER DDB(8,20)
0007        INTEGER TCC
      C
      C
0008        COMMON /DDB/ DDB
      C
0009        COMMON /TC/ TCUIM, TCRIM, TCUGM, TCRGM, TCUPS, TCRPS,
     *                TCUFM, TCRFM, TCUTP, TCRTP, TCUDC, TCRDC,
     *              TCUCB, TCRCB, TCUCD, TCRCD, TCUDB, TCRDB
```

AD-A084 895 MEASUREMENT CONCEPT CORP ROME NY F/G 17/9
TACTICAL TARGET IDENTIFICATION (TTI) LABORATORY SIMULATION. VOL--ETC(U)
MAR 80 R KELLOGG, R A MCKIE, J TIMS F30602-79-C-0029
UNCLASSIFIED RADC -TR-80-41-VOL-1 NL




1.0
1.1
1.25
1.4
1.6
1.8
2.0
2.2
2.5
2.8
3.2
3.6
4.0
4.5
5.0
MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS 1963-A

```
      C
0010  C     COMMON /CHAN/ ICHIN
      C
0011  C     COMMON /COMVAR/ MAXIM,MAXLN,MAXGM
      C
0012  C     COMMON /DATAA/ TCC
      C
      C     END OF DATA DEFINITION
      C
      C     NOTE: ALL CONSTANTS ARE DEFINED IN FUNCTION CMOPEN
      C     ----- ------------                   ------
      C
      C     **********************************************************
      C
      C
      C     1) SET FUNCTION NUMBER
```



```
      C     2) CALL TRANSFER CODE 2 COMMON ROUTINE
      C
0013  C     TCC=TCUPS
      C
0014  C     IUPS=ITC2(IFMNO,BUFF)
      C
0015        RETURN
0016        END
```

RT-11 FORTRAN IV STORAGE MAP

NAME	OFFSET	ATTRIBUTES
BUFF	000014	INTEGER*2 PARAMETER ARRAY (256)
IUPS	000016	INTEGER*2 VARIABLE
ITC2	000000	INTEGER*2 PROCEDURE
IFMNO	000070	INTEGER*2 VARIABLE

COMMON BLOCK /DDB/ LENGTH 000500

DDB	000000	INTEGER*2 ARRAY (8,20) VECTORED

COMMON BLOCK /TC/ LENGTH 000044

TCUIM	000000	INTEGER*2 VARIABLE

```
TCRIM   000002  INTEGER*2 VARIABLE
TCWGM   000004  INTEGER*2 VARIABLE
TCRGM   000006  INTEGER*2 VARIABLE
TCWPS   000010  INTEGER*2 VARIABLE
TCRPS   000012  INTEGER*2 VARIABLE
TCWFM   000014  INTEGER*2 VARIABLE
TCRFM   000016  INTEGER*2 VARIABLE
TCWTP   000020  INTEGER*2 VARIABLE
TCRTP   000022  INTEGER*2 VARIABLE
TCWDC   000024  INTEGER*2 VARIABLE
TCRDC   000026  INTEGER*2 VARIABLE
TCWCB   000030  INTEGER*2 VARIABLE
TCRCB   000032  INTEGER*2 VARIABLE
TCWCD   000034  INTEGER*2 VARIABLE
TCRCD   000036  INTEGER*2 VARIABLE
TCWDB   000040  INTEGER*2 VARIABLE
TCRDB   000042  INTEGER*2 VARIABLE

COMMON BLOCK /CHAN/     LENGTH 000002

ICHIN   000000  INTEGER*2 VARIABLE

COMMON BLOCK /COMVAR/   LENGTH 000006

MAXIM   000000  INTEGER*2 VARIABLE
MAXLN   000002  INTEGER*2 VARIABLE
MAXGM   000004  INTEGER*2 VARIABLE

COMMON BLOCK /DATAA/    LENGTH 000002

TCC     000000  INTEGER*2 VARIABLE
```



```
PT-11 FORTRAN IV        V01B-08A   WED 21-MAR-79 16:28:47       PAGE 001

        C
0001            INTEGER FUNCTION IRPS(BUFF)
        C
        C       FUNCTIONAL DESCRIPTION:
        C
        C       THIS ROUTINE READS THE PSEUDO-COLOR
        C       MEMORY AND PUTS IT IN THE USER BUFFER.
        C
        C       PARAMETERS:
        C
        C       BUFF  -  USER BUFFER (INTEGER ARRAY 256)
        C
        C       ERRORS:
        C
        C        0 -  SUCCESS
        C       -1 -  WRITE  ERROR
        C
```

```
C
C
C     ***********************************************
C
C     DATA DEFINITION
C
0002  C     INTEGER     BUFF(256)
C
C
0003  C     INTEGER TCWIM, TCRIM, TCWGM, TCRGM, TCWPS, TCRPS
0004  C     INTEGER TCWFM, TCRFM, TCWTP, TCRTP, TCWDC, TCRDC
0005  C     INTEGER TCWCB, TCRCB, TCWCD, TCRCD, TCWDB, TCRDB
C
C
0006  C     INTEGER DDB(8,20)
0007  C     INTEGER TCC
C
C
C
0008  C     COMMON /DDB/ DDB
C
0009  C     COMMON /TC/ TCWIM, TCRIM, TCWGM, TCRGM, TCWPS, TCRPS,
     *                  TCWFM, TCRFM, TCWTP, TCRTP, TCWDC, TCRDC,
     *                TCWCB, TCRCB, TCWCD, TCRCD, TCWDB, TCRDB
C
0010  C     COMMON /CHAN/ ICHIN
C
0011  C     COMMON /COMVAR/ MAXIM,MAXLN,MAXGM
C
0012  C     COMMON /DATAA/ TCC
C
C     END OF DATA DEFINITION
C
C     NOTE: ALL CONSTANTS ARE DEFINED IN FUNCTION CMOPEN
C     ----- -------------                    ------
C
C     ***********************************************************
C
C     1) SET FUNCTION NUMBER
```



```
C     2) CALL TRANSFER CODE 2 COMMON ROUTINE
C
0013  C     TCC=TCRPS
C
0014  C     IRPS=ITC2(IFMNO,BUFF)
C
0015  C     RETURN
0016  C     END
```

```
RT-11 FORTRAN IV        STORAGE MAP

NAME    OFFSET  ATTRIBUTES

BUFF    000014  INTEGER*2 PARAMETER ARRAY (256)
IRPS    000016  INTEGER*2 VARIABLE
ITC2    000000  INTEGER*2 PROCEDURE
IFMNO   000070  INTEGER*2 VARIABLE

COMMON BLOCK /DDB/      LENGTH 000500

DDB     000000  INTEGER*2 ARRAY (8,20) VECTORED

COMMON BLOCK /TC/       LENGTH 000044

TCWIM   000000  INTEGER*2 VARIABLE
TCRIM   000002  INTEGER*2 VARIABLE
TCWGM   000004  INTEGER*2 VARIABLE
TCRGM   000006  INTEGER*2 VARIABLE
TCWPS   000010  INTEGER*2 VARIABLE
TCRPS   000012  INTEGER*2 VARIABLE
TCWFM   000014  INTEGER*2 VARIABLE
TCRFM   000016  INTEGER*2 VARIABLE
TCWTP   000020  INTEGER*2 VARIABLE
TCRTP   000022  INTEGER*2 VARIABLE
TCWDC   000024  INTEGER*2 VARIABLE
TCRDC   000026  INTEGER*2 VARIABLE
TCWCB   000030  INTEGER*2 VARIABLE
TCRCB   000032  INTEGER*2 VARIABLE
TCWCD   000034  INTEGER*2 VARIABLE
TCRCD   000036  INTEGER*2 VARIABLE
TCWDB   000040  INTEGER*2 VARIABLE
TCRDB   000042  INTEGER*2 VARIABLE

COMMON BLOCK /CHAN/     LENGTH 000002

ICHIN   000000  INTEGER*2 VARIABLE

COMMON BLOCK /COMVAR/   LENGTH 000006

MAXIM   000000  INTEGER*2 VARIABLE
MAXLN   000002  INTEGER*2 VARIABLE
MAXGM   000004  INTEGER*2 VARIABLE

COMMON BLOCK /DATAA/    LENGTH 000002

TCC     000000  INTEGER*2 VARIABLE
```

```
      C
0001  C     INTEGER FUNCTION ITC2(IFMNO,BUFF)
      C
      C     FUNCTIONAL DESCRIPTION:
      C
      C     THIS ROUTINE IS CALLED BY FUNCTION MEMORY , AND PSEUDO
      C     COLOR MEMORY, READ/WRITE
      C
      C
      C     CALLER SETS THE FUNCTION NUMBER
      C
      C     THE ROUTINE:
      C
      C    1) DETERMINES WHETHER PSEUDO OR FUNCTION (ONLY 1 PSEUDO MEMORY)
      C    2) FORCES FUNCTION MEMORY  < 4
      C    3) SETS COMMAND FOR TRANSFER CODE 2
      C    4) CALLS HANDLER
      C
      C     PARAMETERS:
      C
      C     IFMNO  =  FUNCTION MEMORY NUMBER (1-3)
      C     BUFF   =  USER BUFFER (INTEGER ARRAY 256)
      C
      C     ERRORS:
      C
      C     0 =  SUCCESS
      C    -1 =  WRITE  ERROR
      C
      C
      C    ***********************************************
      C
      C     DATA DEFINITION
      C
0002  C     INTEGER     BUFF(256)
      C
      C
0003  C     INTEGER TCWIM, TCRIM, TCWGM, TCRGM, TCWPS, TCRPS
0004  C     INTEGER TCWFM, TCRFM, TCWTP, TCRTP, TCWDC, TCRDC
0005  C     INTEGER TCWCB, TCRCB, TCWCD, TCRCD, TCWDB, TCRDB
      C
      C
0006  C     INTEGER DDB(8,20)
0007  C     INTEGER TCC
      C
      C
      C
0008  C     COMMON /DDB/ DDB
      C
0009  C     COMMON /TC/ TCWIM, TCRIM, TCWGM, TCRGM, TCWPS, TCRPS,
     2                  TCWFM, TCRFM, TCWTP, TCRTP, TCWDC, TCRDC,
```

```
                *             TCWCB, TCRCB, TCWCD, TCRCD, TCWDB, TCRDB
                C
0010            COMMON /CHAN/ ICHIN
                C
0011            COMMON /COMMAR/ MAXIM,MAXLN,MAXGM
                C
```



```
0012            COMMON /DATAA/ TCC
                C
                C     END OF DATA DEFINITION
                C
                C     NOTE: ALL CONSTANTS ARE DEFINED IN FUNCTION CMOPEN
                C     ----- -------------                    ------
                C
                C     ***************************************************
                C
                C
                C
                C     SET SUCCESS
                C
0013            ITC2=0
                C
                C     CHECK FOR PSEUDO COLOR MEMORY
                C
0014            IF((TCC.EQ.TCWPS).OR.(TCC.EQ.TCRPS)) GO TO 100
                C
                C     FUNCTION MEMORY
                C
                C     CORRECT FUNCTION MEMORY NUMBER
                C     LIMIT FUNCTION MEMORY NUMBER  < 4
                C
0016            NO=IFMNO-1
0017            IF((NO.GT.MAXGM).OR.(NO.LT.0)) NO=0
                C
                C     SET FUNCTION MEMORY NUMBER IN TRANSFER COMMAND
                C
0019            DDB(3,TCC)=NO*400
                C
0020            GO TO 200
                C
                C     PSEUDO COLOR MEMORY
                C
0021  100       DDB(3,TCC)=0
                C
                C     SET BUFFER ADDRESS
                C
0022  200       DDB(4,TCC)=IADDR(BUFF)
                C
                C     CALL HANDLER
```

```
                C
0023            IC=IURITW(256,DDB(1,TCC),1,ICHIN)
                C
                C       CHECK FOR ERROR
                C
0024            IF(IC.LT.0) ITC2=IC
                C
0026            RETURN
0027            END
```

```
RT-11 FORTRAN IV        STORAGE MAP

NAME    OFFSET  ATTRIBUTES

BUFF    000016  INTEGER*2 PARAMETER ARRAY (256)
ITC2    000020  INTEGER*2 VARIABLE
IFMNO   000014  INTEGER*2 PARAMETER VARIABLE
NO      000076  INTEGER*2 VARIABLE
IADDR   000000  INTEGER*2 PROCEDURE
IC      000100  INTEGER*2 VARIABLE
IURITW  000000  INTEGER*2 PROCEDURE

COMMON BLOCK /DDB/      LENGTH 000500

DDB     000000  INTEGER*2 ARRAY (8,20) VECTORED

COMMON BLOCK /TC/       LENGTH 000044

TCWIM   000000  INTEGER*2 VARIABLE
TCRIM   000002  INTEGER*2 VARIABLE
TCWGM   000004  INTEGER*2 VARIABLE
TCRGM   000006  INTEGER*2 VARIABLE
TCWPS   000010  INTEGER*2 VARIABLE
TCRPS   000012  INTEGER*2 VARIABLE
TCWFM   000014  INTEGER*2 VARIABLE
TCPFM   000016  INTEGER*2 VARIABLE
TCWTP   000020  INTEGER*2 VARIABLE
TCRTP   000022  INTEGER*2 VARIABLE
TCWDC   000024  INTEGER*2 VARIABLE
TCRDC   000026  INTEGER*2 VARIABLE
TCWCB   000030  INTEGER*2 VARIABLE
TCRCB   000032  INTEGER*2 VARIABLE
TCWCD   000034  INTEGER*2 VARIABLE
TCPCD   000036  INTEGER*2 VARIABLE
TCWDB   000040  INTEGER*2 VARIABLE
TCRDB   000042  INTEGER*2 VARIABLE

COMMON BLOCK /CHAN/     LENGTH 000002

ICHIN   000000  INTEGER*2 VARIABLE
```

```
COMMON BLOCK /COMVAR/   LENGTH 000006

MAXIM   000000  INTEGER*2 VARIABLE
MAXLN   000002  INTEGER*2 VARIABLE
MAXGM   000004  INTEGER*2 VARIABLE

COMMON BLOCK /DATAA/    LENGTH 000002

TCC     000000  INTEGER*2 VARIABLE
$
```

```
C     FILE NAME: COML04.FOR
C
C     DATE LAST EDITED: 28-FEB-79
C
C
C     HISTORY:
C
C     DATE:                MODIFICATION
C     -----                ------------
C    28-FEB-79    INITAL WRITING
C
C
0001       INTEGER FUNCTION WTP(IX,IY,IXY)
C
C     FUNCTIONAL DESCRIPTION:
C
C     THIS ROUTINE WRITES X, OR Y, OR BOTH, TO THE
C     TARGET CURSOR. (BOTH ACTUALLY REQUIRES 2 TRANSFERS)
C
C     PARAMETERS:
C
C     IX  =  X COORDINATE OF CURSOR (1-256)
C     IY  =  Y COORDINATE OF CURSOR (1-256)
C     IXY
C    =  0   TRANSFER BOTH X AND Y
C    =  1   TRANSFER X ONLY
C    =  2   TRANSFER Y ONLY
C
C     ERRORS:
C
C    0   =        SUCESS
C    -1  =        WRITE ERROR
C
C
C    **************************************************
C     DATA DEFINITION
C
C
0002       INTEGER TCWIM, TCRIM, TCWGM, TCRGM, TCWPS, TCRPS
0003       INTEGER TCWFM, TCRFM, TCWTP, TCRTP, TCWDC, TCRDC
0004       INTEGER TCWCB, TCRCB, TCWCD, TCRCD, TCWDB, TCRDB
C
0005       INTEGER DDB(8,20)
0006       INTEGER TCC
```

```
C
C
0007        COMMON /DDB/ DDB
C
0008        COMMON /TC/ TCUIN, TCRIN, TCUGM, TCRGM, TCUPS, TCRPS,
     $                  TCUFM, TCRFM, TCUTP, TCRTP, TCUDC, TCRDC,
     1              TCUCB, TCRCB, TCUCD, TCRCD, TCUDB, TCRDB
C
0009        COMMON /CHAN/ ICHIN
```



```
C
0010        COMMON /COMVAR/ MAXIN,MAXLN,MAXGM
C
0011        COMMON /DATAA/ TCC
C
C           END OF DATA DEFINITION
C
C           NOTE: ALL CONSTANTS ARE DEFINED IN FUNCTION CMOPEN
C           ----- -------------                    ------
C
C           ***************************************************
C
C
C           SET SUCCESS
C
0012        UTP=0
C
C           SET FUNCTION =URITE TARGET POSITION
C
0013        TCC=TCUTP
C
C           CHECK FOR X,Y,OR BOTH
C
0014        IF(IXY.EQ.0) GO TO 100
0016        IF(IXY.EQ.1) GO TO 200
C
C           JUST Y
C           SET TRANSFER COMMAND
C
0018        DDB(3,TCC)=IY
0019        GO TO 50
C
C
C           JUST X
C
```

```
0020  200   DDB(3,TCC)=IX
0021        GO TO 50
      C
      C
      C     CALL HANDLER
      C
0022  50    IC=IWRITW(256,DDB(1,TCC),1,ICHIN)
      C
      C     CHECK FOR ERROR
      C
0023        IF(IC.LT.0) WTP=IC
0025        RETURN
      C
      C
      C
0026  100   DDB(3,TCC)=IX
0027        IC=IWRITW(256,DDB(1,TCC),256,1,ICHIN)
0028        DDB(3,TCC)=IY
```



```
0029        ID=IWRITW(256,DDB(1,TCC),256,1,ICHIN)
      C
      C     CHECK ERROR
      C
0030        IF(IC.LT.0) WTP=IC
0032        IF(ID.LT.0) WTP=ID
      C
0034        RETURN
0035        END
```

RT-11 FORTRAN IV STORAGE MAP

NAME	OFFSET	ATTRIBUTES
WTP	000022	INTEGER*2 VARIABLE
IX	000014	INTEGER*2 PARAMETER VARIABLE
IY	000016	INTEGER*2 PARAMETER VARIABLE
IXY	000020	INTEGER*2 PARAMETER VARIABLE
IC	000100	INTEGER*2 VARIABLE
IWRITW	000000	INTEGER*2 PROCEDURE
ID	000102	INTEGER*2 VARIABLE

```
COMMON BLOCK /DDB/      LENGTH 000500

DDB     000000  INTEGER*2 ARRAY (8,20) VECTORED

COMMON BLOCK /TC/       LENGTH 000044

TCUIM   000000  INTEGER*2 VARIABLE
TCRIM   000002  INTEGER*2 VARIABLE
TCUGM   000004  INTEGER*2 VARIABLE
TCRGM   000006  INTEGER*2 VARIABLE
TCUPS   000010  INTEGER*2 VARIABLE
TCRPS   000012  INTEGER*2 VARIABLE
TCUFM   000014  INTEGER*2 VARIABLE
TCRFM   000016  INTEGER*2 VARIABLE
TCUTP   000020  INTEGER*2 VARIABLE
TCRTP   000022  INTEGER*2 VARIABLE
TCUDC   000024  INTEGER*2 VARIABLE
TCRDC   000026  INTEGER*2 VARIABLE
TCUCB   000030  INTEGER*2 VARIABLE
TCRCB   000032  INTEGER*2 VARIABLE
TCUCD   000034  INTEGER*2 VARIABLE
TCRCD   000036  INTEGER*2 VARIABLE
TCUDB   000040  INTEGER*2 VARIABLE
TCRDB   000042  INTEGER*2 VARIABLE

COMMON BLOCK /CHAN/     LENGTH 000002

ICHIN   000000  INTEGER*2 VARIABLE

COMMON BLOCK /COMVAR/   LENGTH 000006

MAXIM   000000  INTEGER*2 VARIABLE
MAXLN   000002  INTEGER*2 VARIABLE
MAXGM   000004  INTEGER*2 VARIABLE

COMMON BLOCK /DATAA/    LENGTH 000002

TCC     000000  INTEGER*2 VARIABLE
```



```
      C
      C     INTEGER FUNCTION RTP(IX,IY)
      C
      C     FUNCTIONAL DESCRIPTION:
      C
      C     THIS ROUTINE READS THE X AND Y POSITION OF THE TARGET
      C     AND PUTS THE COORDINATES IN IX, AND IY .
      C
      C     PARAMETERS:
      C
```

```
C     IX  =  X POSITION OF TARGET (1-256)
C     IY  =  Y POSITION OF TARGET (1-256)
C
C     ERRORS:
C
C      0 =  SUCCESS
C     -1 =  WRITE ERROR
C
C     ***********************************************
C
C     DATA DEFINITION
C
0001        INTEGER TAR(2)
C
C
0002        INTEGER TCWIM, TCRIM, TCWGM, TCRGM, TCWPS, TCRPS
0003        INTEGER TCWFM, TCRFM, TCWTP, TCRTP, TCWDC, TCRDC
0004        INTEGER TCWCB, TCRCB, TCWCD, TCRCD, TCWDB, TCRDB
C
C
0005        INTEGER DDB(8,20)
0006        INTEGER TCC
C
C
C
0007        COMMON /DDB/ DDB
C
0008        COMMON /TC/ TCWIM, TCRIM, TCWGM, TCRGM, TCWPS, TCRPS,
           *              TCWFM, TCRFM, TCWTP, TCRTP, TCWDC, TCRDC,
           *           TCWCB, TCRCB, TCWCD, TCRCD, TCWDB, TCRDB
C
0009        COMMON /CHAN/ ICHIN
C
0010        COMMON /COMVAR/ MAXIM,MAXLN,MAXGM
C
0011        COMMON /DATAA/ TCC
C
C     END OF DATA DEFINITION
C
C     NOTE: ALL CONSTANTS ARE DEFINED IN FUNCTION CMOPEN
C     ----- -------------                    ------
C
C     ***********************************************
C
C     SET SUCCESS
C
```



```
0012        RTP=0
```

```
C
C     SET FUNCTION = READ TARGET POSITION
C
0013        TCC=TCRTP
C
C     GET ADDRESS
C
0014        DDB(4,TCC)=IADDR(TAR)
C
C     CALL HANDLER
C
0015        IC=IURITU(256,DDB(1,TCC),1,ICHIN)
C
C     CHECK FOR ERRORS
C
0016        IF(IC.LT.0)RTP=IC
C
C     SET X, AND Y (4097 IS OFFSET FROM 1)
C
0018        IX=TAR(1)+4097
0019        IY=TAR(2)+4097
C
0020        RETURN
0021        END
```



```
RT-11 FORTRAN IV        STORAGE MAP

NAME    OFFSET  ATTRIBUTES

TAR     000006  INTEGER*2 ARRAY (2)
RTP     000066  REAL*4    VARIABLE
IADDR   000000  INTEGER*2 PROCEDURE
IC      000072  INTEGER*2 VARIABLE
IURITU  000000  INTEGER*2 PROCEDURE
IX      000074  INTEGER*2 VARIABLE
IY      000076  INTEGER*2 VARIABLE

COMMON BLOCK /DDB/      LENGTH 000500

DDB     000000  INTEGER*2 ARRAY (8,20) VECTORED

COMMON BLOCK /TC/       LENGTH 000044

TCWIM   000000  INTEGER*2 VARIABLE
TCPIM   000002  INTEGER*2 VARIABLE
TCWGM   000004  INTEGER*2 VARIABLE
TCPGM   000006  INTEGER*2 VARIABLE
TCWPS   000010  INTEGER*2 VARIABLE
TCPPS   000012  INTEGER*2 VARIABLE
TCWFM   000014  INTEGER*2 VARIABLE
TCRFM   000016  INTEGER*2 VARIABLE
```

```
TCUTP    000020  INTEGER*2 VARIABLE
TCRTP    000022  INTEGER*2 VARIABLE
TCUDC    000024  INTEGER*2 VARIABLE
TCRDC    000026  INTEGER*2 VARIABLE
TCWCB    000030  INTEGER*2 VARIABLE
TCRCB    000032  INTEGER*2 VARIABLE
TCWCD    000034  INTEGER*2 VARIABLE
TCRCD    000036  INTEGER*2 VARIABLE
TCUDB    000040  INTEGER*2 VARIABLE
TCRDB    000042  INTEGER*2 VARIABLE

COMMON BLOCK /CHAN/      LENGTH 000002

ICHIN    000000  INTEGER*2 VARIABLE

COMMON BLOCK /COMVAR/    LENGTH 000006

MAXIM    000000  INTEGER*2 VARIABLE
MAXLN    000002  INTEGER*2 VARIABLE
MAXGM    000004  INTEGER*2 VARIABLE

COMMON BLOCK /DATAA/     LENGTH 000002

TCC      000000  INTEGER*2 VARIABLE
*
```

```
C     '                    FILE NAME: COML05.FOR
C     '                    DATE LAST EDITED: 1-MAR-79
C     '
C     '                    HISTORY:
C     '                    DATE:          MODIFICATION
C     '                    -----          ------------
C     '                    1-MAR-79       INITAL WRITING
C
0001        INTEGER FUNCTION IWCB(BUFF)
C           FUNCTIONAL DESCRIPTION:
C
C           THIS ROUTINE IS CALLED TO TRANSMIT A BLOCK OF TEXT
C           TO THE TEXT BUFFER IN THE COMTAL.  AFTER THE TEXT
C           IS RECEIVED, THE COMTAL WILL USE THIS TEXT AS IF IT
C           WERE TYPED FROM THE KEYBOARD.
C
C           PARAMETERS:
C
C           BUFF  -  USER BUFFER OF TEXT
C
C           ERRORS:
C
C            0  -  SUCCESS
C           -1  -  WRITE ERROR
C
C           ***************************************************
C
C           DATA DEFINITION
C
0002        LOGICAL*1 BUFF(512)
C
C
0003        INTEGER TCWIM, TCRIM, TCWGM, TCRGM, TCWPS, TCRPS
0004        INTEGER TCWFM, TCRFM, TCWTP, TCRTP, TCWDC, TCRDC
0005        INTEGER TCWCB, TCRCB, TCWCD, TCRCD, TCWDB, TCRDB
C
C
0006        INTEGER DDB(8,20)
0007        INTEGER TCC
```

```
      C
      C
      C
0005        COMMON /DDB/ DDB
      C
0009        COMMON /TC/ TCUIN, TCRIN, TCUGM, TCRGM, TCUPS, TCRPS,
     $                    TCUFN, TCRFN, TCUTP, TCRTP, TCUDC, TCRDC,
     $                 TCUCB, TCRCB, TCUCD, TCRCD, TCUDB, TCRDB
      C
0010        COMMON /CHAN/ ICHIN
```



```
      C
0011        COMMON /COMVAR/ MAXIM,MAXLN,MAXGM
      C
0012        COMMON /DATAA/ TCC
      C
      C     END OF DATA DEFINITION
      C
      C     NOTE: ALL CONSTANTS ARE DEFINED IN FUNCTION CMOPEN
      C     ----- -------------                              ------
      C
      C     ***************************************************
      C
      C
      C     ,    1) SET COMMAND
      C     ,    2) CALL COMMON CODE FOR TRANSFER CODE 3 TYPE COMMANDS
      C
0013        TCC=TCUCB
      C
0014        IUCB=ITC3(BUFF,IADR)
      C
0015        RETURN
0016        END
```


FT-11 FORTRAN IV STORAGE MAP

NAME	OFFSET	ATTRIBUTES
BUFF	000014	LOGICAL*1 PARAMETER ARRAY (512)
IUCB	000016	INTEGER*2 VARIABLE
ITC3	000000	INTEGER*2 PROCEDURE
IADR	000070	INTEGER*2 VARIABLE

```
COMMON BLOCK  DDB       LENGTH 000500

DDB      000000  INTEGER*2 ARRAY (8,20) VECTORED

COMMON BLOCK /TC/       LENGTH 000044

TCUIN    000000  INTEGER*2 VARIABLE
TCRIM    000002  INTEGER*2 VARIABLE
TCUGM    000004  INTEGER*2 VARIABLE
TCRGM    000006  INTEGER*2 VARIABLE
TCUPS    000010  INTEGER*2 VARIABLE
TCRPS    000012  INTEGER*2 VARIABLE
TCUFM    000014  INTEGER*2 VARIABLE
TCRFM    000016  INTEGER*2 VARIABLE
TCUTP    000020  INTEGER*2 VARIABLE
TCRTP    000022  INTEGER*2 VARIABLE
TCUDC    000024  INTEGER*2 VARIABLE
TCRDC    000026  INTEGER*2 VARIABLE
TCUCB    000030  INTEGER*2 VARIABLE
TCRCB    000032  INTEGER*2 VARIABLE
TCUCD    000034  INTEGER*2 VARIABLE
TCRCD    000036  INTEGER*2 VARIABLE
TCUDB    000040  INTEGER*2 VARIABLE
TCRDB    000042  INTEGER*2 VARIABLE

COMMON BLOCK /CHAN/     LENGTH 000002

ICHIN    000000  INTEGER*2 VARIABLE

COMMON BLOCK /COMVAR/   LENGTH 000006

MAXIM    000000  INTEGER*2 VARIABLE
MAXLN    000002  INTEGER*2 VARIABLE
MAXGM    000004  INTEGER*2 VARIABLE

COMMON BLOCK /DATAA/    LENGTH 000002

TCC      000000  INTEGER*2 VARIABLE
```



```
      C
000:        INTEGER FUNCTION IRCB(BUFF)
      C
      C     FUNCTIONAL DESCRIPTION:
      C
      C     THIS ROUTINE IS CALLED TO READ THE BLOCK OF TEXT
      C     FROM THE BUFFER IN THE COMTAL.  THIS WOULD BE USED
      C     WHEN THE COMTAL INTERUPTED VIA THE 'SEND MESSAGE' REQUEST,
      C     TO READ THE TEXT TO DETERMINE WHAT THE MESSGE WAS.
```

```
C
C     PARAMETERS:
C
C     BUFF  -  USER BUFFER FOR TEXT
C
C     ERRORS:
C
C      0  -  SUCCESS
C     -1  -  WRITE ERROR
C
C
C     ***************************************************
C
C     DATA DEFINITION
C
0002  C     LOGICAL*1 BUFF(512)
C     C
C     C
0003  C     INTEGER TCWIM, TCRIM, TCWGM, TCRGM, TCWPS, TCRPS
0004  C     INTEGER TCWFM, TCRFM, TCWTP, TCRTP, TCWDC, TCRDC
0005  C     INTEGER TCWCB, TCRCB, TCWCD, TCRCD, TCWDB, TCRDB
C     C
C     C
0006  C     INTEGER DDB(8,20)
0007  C     INTEGER TCC
C     C
C     C
0008  C     COMMON /DDB/ DDB
C     C
0009  C     COMMON /TC/ TCWIM, TCRIM, TCWGM, TCRGM, TCWPS, TCRPS,
     1                  TCWFM, TCRFM, TCWTP, TCRTP, TCWDC, TCRDC,
     2                  TCWCB, TCRCB, TCWCD, TCRCD, TCWDB, TCRDB
C     C
0010  C     COMMON /CHAN/ ICHIN
C     C
0011  C     COMMON /COMVAR/ MAXIM,MAXLN,MAXGM
C     C
0012  C     COMMON /DATAA/ TCC
C     C
C     C     END OF DATA DEFINITION
C     C
C     C     NOTE: ALL CONSTANTS ARE DEFINED IN FUNCTION CMOPEN
C     C     ----- -------------                    ------
C     C
C     C     ***************************************************
```

```
C
C     1) SET COMMAND
C     2) CALL COMMON CODE FOR TRANSFER CODE 3 TYPE COMMANDS
0013        TCC=TCRCB
C
0014        IRCB=ITC3(BUFF,IADR)
C
0015        RETURN
0016        END
```

```
RT-11 FORTRAN IV        STORAGE MAP

NAME    OFFSET  ATTRIBUTES

BUFF    000014  LOGICAL*1 PARAMETER ARRAY (512)
IRCB    000016  INTEGER*2 VARIABLE
ITC3    000000  INTEGER*2 PROCEDURE
IADR    000070  INTEGER*2 VARIABLE

COMMON BLOCK /DDB/      LENGTH 000500

DDB     000000  INTEGER*2 ARRAY (8,20) VECTORED

COMMON BLOCK /TC/       LENGTH 000044

TCUIM   000000  INTEGER*2 VARIABLE
TCRIM   000002  INTEGER*2 VARIABLE
TCUGM   000004  INTEGER*2 VARIABLE
TCRGM   000006  INTEGER*2 VARIABLE
TCUPS   000010  INTEGER*2 VARIABLE
TCRPS   000012  INTEGER*2 VARIABLE
TCUFM   000014  INTEGER*2 VARIABLE
TCRFM   000016  INTEGER*2 VARIABLE
TCUTP   000020  INTEGER*2 VARIABLE
TCRTP   000022  INTEGER*2 VARIABLE
TCWDC   000024  INTEGER*2 VARIABLE
TCRDC   000026  INTEGER*2 VARIABLE
TCUCB   000030  INTEGER*2 VARIABLE
TCRCB   000032  INTEGER*2 VARIABLE
TCWCD   000034  INTEGER*2 VARIABLE
TCRCD   000036  INTEGER*2 VARIABLE
TCWDB   000040  INTEGER*2 VARIABLE
TCRDB   000042  INTEGER*2 VARIABLE

COMMON BLOCK /CHAN/     LENGTH 000002

ICHIN   000000  INTEGER*2 VARIABLE

COMMON BLOCK /COMVAR/   LENGTH 000006
```

```
M4\IM   000000  INTEGER*2 VARIABLE
M4.LN   000002  INTEGER*2 VARIABLE
M4\GM   000004  INTEGER*2 VARIABLE

COMMON BLOCK  DATAA     LENGTH 000002

TCC     000000  INTEGER*2 VARIABLE



RT-11 FORTRAN IV        V01B-08A   WED 18-APR-79 16:30:13       PAGE 001

        C
0001            INTEGER FUNCTION UDB(BUFF)
        C
        C       FUNCTIONAL DESCRIPTION:
        C
        C       THIS ROUTINE IS CALLED TO TRANSMIT THE DATA BLOCK
        C       TO THE COMTAL.  THE DATA BLOCK CONTAINS THE CONTENTS
        C       OF SEVERAL DISPLAY REGISTERS THAT DETERMINE THE STATUS
        C       OF THE DISPLAY.
        C
        C
        C       PARAMETERS:
        C
        C       BUFF  =  USER BUFFER OF TEXT
        C
        C       ERRORS:
        C
        C        0  =  SUCCESS
        C       -1  =  WRITE ERROR
        C
        C
        C       ***********************************************
        C
        C       DATA DEFINITION
        C
0002            LOGICAL*1 BUFF(512)
        C
        C
0003            INTEGER TCWIM, TCRIM, TCWGM, TCRGM, TCWPS, TCRPS
0004            INTEGER TCWFM, TCRFM, TCWTP, TCRTP, TCWDC, TCRDC
0005            INTEGER TCWCB, TCRCB, TCWCD, TCRCD, TCWDB, TCRDB
        C
        C
0006            INTEGER DDB(8,20)
0007            INTEGER TCC
        C
        C
        C
0008            COMMON /DDB/ DDB
```

```
0005      C     COMMON /TC/ TCUIM, TCRIM, TCUGM, TCRGM, TCUPS, TCRPS,
     &                    TCUFM, TCRFM, TCUTP, TCRTP, TCUDC, TCRDC,
     &                    TCUCB, TCRCB, TCUCD, TCRCD, TCUDB, TCRDB
0010      C     COMMON /CHAN/ ICHIN
0011      C     COMMON /COMVAR/ MAXIM,MAXLN,MAXGM
0012      C     COMMON /DATAA/ TCC
          C
          C     END OF DATA DEFINITION
          C
          C     NOTE: ALL CONSTANTS ARE DEFINED IN FUNCTION CMOPEN
          C     ----- -------------                        ------
          C
          C     ***************************************************
```



```
          C
          C     1) SET COMMAND
          C     2) CALL COMMON CODE FOR TRANSFER CODE 3 TYPE COMMANDS
          C
0013           TCC=TCUDB
          C
0014           UDB=ITC3(BUFF,IADR)
          C
0015           RETURN
0016           END
```

RT-11 FORTRAN IV STORAGE MAP

NAME	OFFSET	ATTRIBUTES
BUFF	000014	LOGICAL*1 PARAMETER ARRAY (512)
UDB	000016	INTEGER*2 VARIABLE
ITC3	000000	INTEGER*2 PROCEDURE
IADR	000070	INTEGER*2 VARIABLE

COMMON BLOCK /DDB/ LENGTH 000500

DDB	000000	INTEGER*2 ARRAY (8,20) VECTORED

```
COMMON BLOCK /TC/       LENGTH 000044

TCWIN   000000  INTEGER*2 VARIABLE
TCRIN   000002  INTEGER*2 VARIABLE
TCWGN   000004  INTEGER*2 VARIABLE
TCRGN   000006  INTEGER*2 VARIABLE
TCWPS   000010  INTEGER*2 VARIABLE
TCRPS   000012  INTEGER*2 VARIABLE
TCWFN   000014  INTEGER*2 VARIABLE
TCRFN   000016  INTEGER*2 VARIABLE
TCWTP   000020  INTEGER*2 VARIABLE
TCRTP   000022  INTEGER*2 VARIABLE
TCWDC   000024  INTEGER*2 VARIABLE
TCRDC   000026  INTEGER*2 VARIABLE
TCWCB   000030  INTEGER*2 VARIABLE
TCRCB   000032  INTEGER*2 VARIABLE
TCWCD   000034  INTEGER*2 VARIABLE
TCRCD   000036  INTEGER*2 VARIABLE
TCWDB   000040  INTEGER*2 VARIABLE
TCRDB   000042  INTEGER*2 VARIABLE

COMMON BLOCK /CHAN/     LENGTH 000002

ICHIN   000000  INTEGER*2 VARIABLE

COMMON BLOCK /COMVAR/   LENGTH 000006

MAXIN   000000  INTEGER*2 VARIABLE
MAXLN   000002  INTEGER*2 VARIABLE
MAXGN   000004  INTEGER*2 VARIABLE

COMMON BLOCK /DATAA/    LENGTH 000002

TCC     000000  INTEGER*2 VARIABLE
```



```
RT-11 FORTRAN IV        V01B-08A   WED 18-APR-79 16:30:18       PAGE 001

      C
0001        INTEGER FUNCTION RDB(BUFF)
      C
      C     FUNCTIONAL DESCRIPTION:
      C
      C     THIS ROUTINE IS CALLED TO READ THE COMTAL DATA BLOCK.
      C     THE DATA BLOCK CONTAINS THE TARGET POSITION, AND THE
      C     DISPLAY REGISTERS.
      C
      C
      C
      C     PARAMETERS:
      C
```

```
C     BUFF  -  USER BUFFER OF TEXT
C
C     ERRORS:
C
C      0  -  SUCCESS
C     -1  -  WRITE ERROR
C
C
C     ***********************************************
C
C     DATA DEFINITION
C
0002        LOGICAL*1 BUFF(512)
C
C
0003        INTEGER TCWIM, TCRIM, TCWGM, TCRGM, TCWPS, TCRPS
0004        INTEGER TCWFM, TCRFM, TCWTP, TCRTP, TCWDC, TCRDC
0005        INTEGER TCWCB, TCRCB, TCWCD, TCRCD, TCWDB, TCRDB
C
C
0006        INTEGER DDB(8,20)
0007        INTEGER TCC
C
C
C
0008        COMMON /DDB/ DDB
C
0009        COMMON /TC/ TCWIM, TCRIM, TCWGM, TCRGM, TCWPS, TCRPS,
     *                  TCWFM, TCRFM, TCWTP, TCRTP, TCWDC, TCRDC,
     *                TCWCB, TCRCB, TCWCD, TCRCD, TCWDB, TCRDB
C
0010        COMMON /CHAN/ ICHIN
C
0011        COMMON /COMVAR/ MAXIM,MAXLN,MAXGM
C
0012        COMMON /DATAA/ TCC
C
C     END OF DATA DEFINITION
C
C     NOTE: ALL CONSTANTS ARE DEFINED IN FUNCTION CMOPEN
C     ----- -------------                       ------
C
C     ***********************************************************
```



```
C
C
C     1) SET COMMAND
C     2) CALL COMMON CODE FOR TRANSFER CODE 3 TYPE COMMANDS
```

```
C
      TCC=TCRDB
C
      RDB=ITC3(BUFF,IADR)
C
      RETURN
      END
```

```
RT-11 FORTRAN IV        STORAGE MAP

NAME    OFFSET  ATTRIBUTES

BUFF    000014  LOGICAL*1 PARAMETER ARRAY (512)
RDB     000016  INTEGER*2 VARIABLE
ITC3    000000  INTEGER*2 PROCEDURE
IADR    000070  INTEGER*2 VARIABLE

COMMON BLOCK /DDB/      LENGTH 000500

DDB     000000  INTEGER*2 ARRAY (8,20) VECTORED

COMMON BLOCK /TC/       LENGTH 000044

TCUIM   000000  INTEGER*2 VARIABLE
TCRIM   000002  INTEGER*2 VARIABLE
TCUGM   000004  INTEGER*2 VARIABLE
TCRGM   000006  INTEGER*2 VARIABLE
TCUPS   000010  INTEGER*2 VARIABLE
TCRPS   000012  INTEGER*2 VARIABLE
TCUFM   000014  INTEGER*2 VARIABLE
TCRFM   000016  INTEGER*2 VARIABLE
TCUTP   000020  INTEGER*2 VARIABLE
TCRTP   000022  INTEGER*2 VARIABLE
TCUDC   000024  INTEGER*2 VARIABLE
TCPDC   000026  INTEGER*2 VARIABLE
TCUCB   000030  INTEGER*2 VARIABLE
TCRCB   000032  INTEGER*2 VARIABLE
TCUCD   000034  INTEGER*2 VARIABLE
TCRCD   000036  INTEGER*2 VARIABLE
TCUDB   000040  INTEGER*2 VARIABLE
TCRDB   000042  INTEGER*2 VARIABLE

COMMON BLOCK /CHAN/     LENGTH 000002

ICHIN   000000  INTEGER*2 VARIABLE

COMMON BLOCK /COMVAR/   LENGTH 000006

MAXIM   000000  INTEGER*2 VARIABLE
MAXLN   000002  INTEGER*2 VARIABLE
MAXGM   000004  INTEGER*2 VARIABLE
```

```
COMMON BLOCK /DATAA/    LENGTH 000002

TCC     000000  INTEGER*2 VARIABLE



RT-11 FORTRAN IV        V01B-08A   WED 18-APR-79 16:30:22       PAGE 001

                C
0001                INTEGER FUNCTION IWCD(BUFF,IAD)
                C
                C   FUNCTIONAL DESCRIPTION:
                C
                C   THIS ROUTINE IS USED TO WRITE TO ANY BLOCK OF MEMORY
                C   IN THE COMTAL.  ITS PRIMARY USE IS IN DOWN LOADING CODE
                C   TO THE COMTAL, HOWEVER ANY OF THE COMTAL MEMORY LOC.
                C   CAN BE READ, AND ANY OF THE RAM AREAS WRITTEN.
                C   THE ADDRESS MUST BE ON A 256 WORD BOUNDARY.
                C
                C   PARAMETERS:
                C
                C   BUFF  -  USER BUFFER OF TEXT
                C   IAD   -  ADDRESS IN THE COMTAL  FOR THE TRANSFER
                C  (MUST BE A 512 BYTE BOUNDARY)
                C
                C   ERRORS:
                C
                C    0  -  SUCCESS
                C   -1  -  WRITE ERROR
                C
                C
                C  **********************************************
                C
                C   DATA DEFINITION
                C
0002                LOGICAL*1 BUFF(512)
                C
                C
0003                INTEGER TCWIM, TCRIM, TCWGM, TCRGM, TCWPS, TCRPS
0004                INTEGER TCWFM, TCRFM, TCWTP, TCRTP, TCWDC, TCRDC
0005                INTEGER TCWCB, TCRCB, TCWCD, TCRCD, TCWDB, TCRDB
                C
                C
0006                INTEGER DDB(8,20)
0007                INTEGER TCC
                C
                C
                C
0008                COMMON /DDB/ DDB
                C
```

```
0009          COMMON /TC/ TCWIM, TCRIM, TCWGM, TCRGM, TCWPS, TCRPS,
     $                    TCWFM, TCRFM, TCWTP, TCRTP, TCWDC, TCRDC,
     $                 TCWCB, TCRCB, TCWCD, TCRCD, TCWDB, TCRDB
C
0010          COMMON /CHAN/ ICHIN
C
0011          COMMON /COMVAR/ MAXIM,MAXLN,MAXGM
C
0012          COMMON /DATAA/ TCC
C
C             END OF DATA DEFINITION
C
C             NOTE: ALL CONSTANTS ARE DEFINED IN FUNCTION CMOPEN
```



```
C     ----- -------------                              ------
C
C     ***************************************************
C
C
C     1) SET COMMAND
C     2) CALL COMMON CODE FOR TRANSFER CODE 3 TYPE COMMANDS
C
0013          TCC=TCWCD
C
0014          IWCD=ITC3(BUFF,IADR)
C
0015          RETURN
0016          END
```


RT-11 FORTRAN IV STORAGE MAP

NAME	OFFSET	ATTRIBUTES
BUFF	000014	LOGICAL*1 PARAMETER ARRAY (512)
IWCD	000020	INTEGER*2 VARIABLE
IAD	000016	INTEGER*2 PARAMETER VARIABLE
ITC3	000000	INTEGER*2 PROCEDURE
IADR	000072	INTEGER*2 VARIABLE

COMMON BLOCK /DDB/ LENGTH 000500

DDB	000000	INTEGER*2 ARRAY (8,20) VECTORED

```
COMMON BLOCK /TC/       LENGTH 000044

TCUIM   000000  INTEGER*2 VARIABLE
TCRIM   000002  INTEGER*2 VARIABLE
TCUGM   000004  INTEGER*2 VARIABLE
TCRGM   000006  INTEGER*2 VARIABLE
TCUPS   000010  INTEGER*2 VARIABLE
TCRPS   000012  INTEGER*2 VARIABLE
TCUFM   000014  INTEGER*2 VARIABLE
TCRFM   000016  INTEGER*2 VARIABLE
TCUTP   000020  INTEGER*2 VARIABLE
TCRTP   000022  INTEGER*2 VARIABLE
TCUDC   000024  INTEGER*2 VARIABLE
TCRDC   000026  INTEGER*2 VARIABLE
TCUCB   000030  INTEGER*2 VARIABLE
TCRCB   000032  INTEGER*2 VARIABLE
TCUCD   000034  INTEGER*2 VARIABLE
TCRCD   000036  INTEGER*2 VARIABLE
TCUDB   000040  INTEGER*2 VARIABLE
TCRDB   000042  INTEGER*2 VARIABLE

COMMON BLOCK /CHAN/     LENGTH 000002

ICHIN   000000  INTEGER*2 VARIABLE

COMMON BLOCK /COMVAR/   LENGTH 000006

MAXIM   000000  INTEGER*2 VARIABLE
MAXLN   000002  INTEGER*2 VARIABLE
MAXGM   000004  INTEGER*2 VARIABLE

COMMON BLOCK /DATAA/    LENGTH 000002

TCC     000000  INTEGER*2 VARIABLE
```



```
PT-11 FORTRAN IV        V01B-08A   WED 18-APR-79 16:30:26       PAGE 001

        C
0001            INTEGER FUNCTION IRCD(BUFF,IAD)
        C
        C       FUNCTIONAL DESCRIPTION:
        C
        C       THIS ROUTINE IS USED TO READ FROM ANY BLOCK OF MEMORY
        C       IN THE COMTAL.   THE ADDRESS MUST BE ON A 256 WORD
        C       BOUNDARY.
        C
        C
        C
        C       PARAMETERS:
        C
```

```
C     BUFF  =  USER BUFFER OF TEXT
C     IAD   =  ADDRESS IN THE CONTAL  FOR THE TRANSFER
C
C     ERRORS:
C
C      0  =  SUCCESS
C     -1  =  WRITE ERROR
C
C     ***********************************************
C     DATA DEFINITION
C
0002  C     LOGICAL*1 BUFF(512)
C
0003  C     INTEGER TCWIM, TCRIM, TCWGM, TCRGM, TCWPS, TCRPS
0004  C     INTEGER TCWFM, TCRFM, TCWTP, TCRTP, TCWDC, TCRDC
0005  C     INTEGER TCWCB, TCRCB, TCWCD, TCRCD, TCWDB, TCRDB
C
0006  C     INTEGER DDB(8,20)
0007  C     INTEGER TCC
C
C
0008  C     COMMON /DDB/ DDB
C
0009  C     COMMON /TC/ TCWIM, TCRIM, TCWGM, TCRGM, TCWPS, TCRPS,
     *                  TCWFM, TCRFM, TCWTP, TCRTP, TCWDC, TCRDC,
     *                  TCWCB, TCRCB, TCWCD, TCRCD, TCWDB, TCRDB
C
0010  C     COMMON /CHAN/ ICHIN
C
0011  C     COMMON /COMAR/ MAXIM,MAXLN,MAXGM
C
0012  C     COMMON /DATAA/ TCC
C
C     END OF DATA DEFINITION
C
C     NOTE: ALL CONSTANTS ARE DEFINED IN FUNCTION CMOPEN
C     ----- -------------                          ------
C
```



```
C     ***********************************************************
C
C     1) SET COMMAND
```

```
C     2) CALL COMMON CODE FOR TRANSFER CODE 3 TYPE COMMANDS
C
0013        TCC=TCRCD
C
0014        IRCD=ITC3(BUFF,IADR)
C
0015        RETURN
0016        END
```

```
PT-11 FORTRAN IV        STORAGE MAP

NAME    OFFSET  ATTRIBUTES

BUFF    000014  LOGICAL*1 PARAMETER ARRAY (512)
IRCD    000020  INTEGER*2 VARIABLE
IAD     000016  INTEGER*2 PARAMETER VARIABLE
ITC3    000000  INTEGER*2 PROCEDURE
IADR    000072  INTEGER*2 VARIABLE

COMMON BLOCK /DDB/      LENGTH 000500

DDB     000000  INTEGER*2 ARRAY (8,20) VECTORED

COMMON BLOCK /TC/       LENGTH 000044

TCWIM   000000  INTEGER*2 VARIABLE
TCRIM   000002  INTEGER*2 VARIABLE
TCWGM   000004  INTEGER*2 VARIABLE
TCRGM   000006  INTEGER*2 VARIABLE
TCWPS   000010  INTEGER*2 VARIABLE
TCRPS   000012  INTEGER*2 VARIABLE
TCWFM   000014  INTEGER*2 VARIABLE
TCRFM   000016  INTEGER*2 VARIABLE
TCWTP   000020  INTEGER*2 VARIABLE
TCRTP   000022  INTEGER*2 VARIABLE
TCWDC   000024  INTEGER*2 VARIABLE
TCRDC   000026  INTEGER*2 VARIABLE
TCWCB   000030  INTEGER*2 VARIABLE
TCRCB   000032  INTEGER*2 VARIABLE
TCWCD   000034  INTEGER*2 VARIABLE
TCRCD   000036  INTEGER*2 VARIABLE
TCWDB   000040  INTEGER*2 VARIABLE
TCRDB   000042  INTEGER*2 VARIABLE

COMMON BLOCK /CHAN/     LENGTH 000002

ICHIN   000000  INTEGER*2 VARIABLE

COMMON BLOCK /COMVAR/   LENGTH 000006

MAXIM   000000  INTEGER*2 VARIABLE
```

```
MAXLN   000002  INTEGER*2 VARIABLE
MAXGM   000004  INTEGER*2 VARIABLE

COMMON BLOCK /DATAA/    LENGTH 000002

TCC     000000  INTEGER*2 VARIABLE
```



```
      C
      C
0001        INTEGER FUNCTION ITC3(BUFF,IAD)
      C
      C     FUNCTIONAL DESCRIPTION:
      C
      C     THIS IS THE COMMON ROUTINE FOR TRANSFER CODE 3
      C     TYPE COMMANDS.  IT IS CALLED BY, READ/WRITE - COMMAND
      C     BLOCK, CODE DATA, AND DATA BLOCK.
      C
      C     PARAMETERS:
      C
      C     BUFF  -  USER BUFFER OF TEXT
      C     IAD   -  ADDRESS IN THE CONTAL  FOR THE TRANSFER
      C
      C     ERRORS:
      C
      C      0  -  SUCCESS
      C     -1  -  WRITE ERROR
      C
      C
      C     **********************************************
      C
      C     DATA DEFINITION
      C
0002        LOGICAL*1 BUFF(512)
      C
      C
0003        INTEGER TCWIM, TCRIM, TCWGM, TCRGM, TCUPS, TCRPS
0004        INTEGER TCWFM, TCRFM, TCUTP, TCRTP, TCUDC, TCRDC
0005        INTEGER TCWCB, TCRCB, TCWCD, TCRCD, TCUDB, TCRDB
      C
      C
0006        INTEGER DDB(8,20)
0007        INTEGER TCC
      C
      C
      C
0008        COMMON /DDB/ DDB
      C
```

```
0009          COMMON /TC/ TCUIM, TCRIM, TCUGM, TCRGM, TCUPS, TCRPS,
     *                    TCUFM, TCRFM, TCUTP, TCRTP, TCUDC, TCRDC,
     *                 TCUCB, TCRCB, TCUCD, TCRCD, TCUDB, TCRDB
              C
0010          COMMON /CHAN/ ICHIN
              C
0011          COMMON /COMUAR/ MAXIM,MAXLN,MAXGM
              C
0012          COMMON /DATAA/ TCC
              C
              C     END OF DATA DEFINITION
              C
              C     NOTE: ALL CONSTANTS ARE DEFINED IN FUNCTION CMOPEN
              C     ----- -------------                              ------
              C
```



```
              C     **********************************************************
              C
              C
0013          IA=IAD
              C
              C     SET SUCCESS
              C
0014          ITC3=0
              C
              C     GET ADDRESS
              C
0015          DDB(4,TCC)=IADDR(BUFF)
              C
0016          DDB(3,TCC)=0
              C
              C     CHECK FOR READ/WRITE OF CODE
              C
0017          IF((TCC.NE.TCWCD).AND.(TCC.NE.TCRCD)) GO TO 100
              C
              C     FUNCTION = CODE DATA, SET TRANSFER COMMAND
              C     FORCE TO EVEN 256 WORD PAGE
              C
0019          DDB(3,TCC)=(IAD/256)*256
              C
              C     CALL HANDLER
              C
0020  100     IC=IWRITW(256,DDB(1,TCC),1,ICHIN)
0021          IF(IC.LT.0)ITC3=IC
0023          RETURN
0024          END
```

```
RT-11 FORTRAN IV        STORAGE MAP

NAME    OFFSET  ATTRIBUTES

BUFF    000014  LOGICAL*1 PARAMETER ARRAY (512)
ITC3    000020  INTEGER*2 VARIABLE
IAD     000016  INTEGER*2 PARAMETER VARIABLE
IA      000076  INTEGER*2 VARIABLE
IADDR   000000  INTEGER*2 PROCEDURE
IC      000100  INTEGER*2 VARIABLE
IWRITW  000000  INTEGER*2 PROCEDURE

COMMON BLOCK /DDB/      LENGTH 000500

DDB     000000  INTEGER*2 ARRAY (8,20) VECTORED

COMMON BLOCK /TC/       LENGTH 000044

TCWIM   000000  INTEGER*2 VARIABLE
TCRIM   000002  INTEGER*2 VARIABLE
TCWGM   000004  INTEGER*2 VARIABLE
TCRGM   000006  INTEGER*2 VARIABLE
TCWPS   000010  INTEGER*2 VARIABLE
TCRPS   000012  INTEGER*2 VARIABLE
TCWFM   000014  INTEGER*2 VARIABLE
TCRFM   000016  INTEGER*2 VARIABLE
TCWTP   000020  INTEGER*2 VARIABLE
TCRTP   000022  INTEGER*2 VARIABLE
TCWDC   000024  INTEGER*2 VARIABLE
TCRDC   000026  INTEGER*2 VARIABLE
TCWCB   000030  INTEGER*2 VARIABLE
TCRCB   000032  INTEGER*2 VARIABLE
TCWCD   000034  INTEGER*2 VARIABLE
TCRCD   000036  INTEGER*2 VARIABLE
TCWDB   000040  INTEGER*2 VARIABLE
TCRDB   000042  INTEGER*2 VARIABLE

COMMON BLOCK /CHAN/     LENGTH 000002

ICHIN   000000  INTEGER*2 VARIABLE

COMMON BLOCK /COMVAR/   LENGTH 000006

MAXIM   000000  INTEGER*2 VARIABLE
MAXLN   000002  INTEGER*2 VARIABLE
MAXGM   000004  INTEGER*2 VARIABLE

COMMON BLOCK /DATAA/    LENGTH 000002

TCC     000000  INTEGER*2 VARIABLE
*
```

2.0 INTEGRATION OF THE COMTAL VISION ONE WITH THE RIPS (RADAR) IMAGE PROCESSING SYSTEM

The PAR RIPS system was modified to incorporate a new option "frame". which allows a RIPS user to write and read information in RIPS format to and from the COMTAL Vision One display system. Mechanisms for adding functions to the RIPS system had been built in and documented by PAR, simplifying this effort by Mc^2.

The approach taken was to modify the COMTAL communication facilities provided by Mc^2's previous COMPIP subsystem to run under RSX-11M, and then to integrate this package with RIPS. Implicit in this effort was the thorough checkout of the COMTAL-supplied RSX-11M device driver, purchased under this contract. This handler is essentially a DEC DMA driver, modified by COMTAL to support their hardware.

In addition to adding the COMTAL to RIPS as a valid input/output device, this sub-effort has produced a set of routines that can be used by any FORTRAN programmer for communicating with the COMTAL under RSX-11M. Evidence of this is the effort of Section III following, where the SRC SILHOUETTE program, written in FORTRAN under RT-11, was modified to use the COMTAL under RSX-11M.

Instructions for altering the RIPS subsystem to accommodate another "menu" of user options were included in PAR's RIPS documentation. Stated briefly, these steps are:

- Update RIPS file FRAMES.SRC, which contains the text and control transfer codes of the screen menus appearing in RIPS.
- Run the FRAMES program to convert the above file into its RIPS-usable form, FRAMES.IPS.
- Add control transfer code to RIPS module CALSUB. Re-compile CALSUB.

- Update the Overlay Descriptor Language File EXEC.ODL to incorporate any new Mc^2 software being added to RIPS.
- Compile or assemble new code to be incorporated in the modified RIPS system.
- Run the indirect command file RIPSLNK to create the new version of RIPS.

The accompanying listings show the code either written or modified by Mc^2 to add the COMTAL to RIPS.

2.1 FRAMES.SRC and FRAMES.IPS

Modifications:

A new RIPS "FRAME" or option menu was added to FRAMES.SRC to allow user selection of one of the nine COMTAL read/write functions added to RIPS. The following excerpt from FRAMES.SRC is annotated to indicate the code added by Mc^2.

```
.
.
.
12
F
0
UTILITY FUNCTIONS$
RSX$
LOG$
CALL USER PROGRAM$
GENETATE TEST IMAGE$
CREATE TEST IMAGE$
EXAMINE HEADER TEXT$
LIST IMAGE GREY VALUES$
DISPLAY OPTIONS$
MASTER FRAME$
11
0
1
0
2
0
3
0
4
0
5
0
6
0
7
0
8
0
9
F
12
F
0
COMTAL DISPLAY AND COPY OPTIONS$
DISPLAY IMAGE FILE$
COPY IMAGE FILE FROM COMTAL$
DISPLAY GRAPHICS FILE$
COPY GRAPHICS FILE FROM COMTAL$
DISPLAY FUNCTION FILE$
COPY FUNCTION FILE FROM COMTAL$
DISPLAY PSEUDOCOLOR FILE$
COPY PSEUDOCOLOR FILE FROM COMTAL$
WRITE COMMAND BLOCK TO COMTAL$
DISPLAY OPTIONS$
MASTER FRAME$
```

Added by Mc^2

(end of FRAMES.SRC)

2.2 CALSUB.FTN

Modifications:

- Computed "GO TO" statement, which is first executable statement of CALSUB, has the argument after 13 changed from a 21 (signifying option not implemented) to a 14.
- Code beginning one line before statement 14 and ending at the RETURN statement is added. These lines call the proper Mc^2 - written routine corresponding to the COMTAL option selected by the user from frame 14.

2.3 EXEC.ODL

Modifications:

- Lines L34 - L39 are added, defining the residency requirements of all new code added by Mc^2 to support the COMTAL.

2.4 CPIPSB.FTN

Authors:

Written for RT-11 by R. Kellogg, Measurement Concept Corp., 1979.

Adapted for RSX-11M by J. Tims, Measurement Concept Corp., 1979.

CPIPSB can be thought of as the interface between Mc^2 software and the RIPS environment. CPIPSB provides the RIPS system access to Mc^2 software, while at the same time, insuring that all requests for normal RIPS I/O services related to the COMTAL requests are passed through RIPS and RIPS's logging services. It is in actuality, a RIPS subroutine, written and added by Mc^2. Other Mc^2 COMTAL software

is independent of all RIPS conventions.

CPIPSB contains nine subroutines, one corresponding to each of the RIPS frame 14 COMTAL options.

2.5 COMLØ1.FOR

Authors:

Written for RT-11 by R. Kellogg, Measurement Concept Corp., 1979.
Modified for RSX-11M by J. Tims, Measurement Concept Corp., 1979.

COMLØ1 is a data definition block with no executable statements. It provides data element definition information, data equivalence information, and COMTAL function code arrays for passing to IWRITW, which performs the actual RSX-11M QIO$ operation.

Information in COMLØ1 is used by COMLØ2, COMLØ3, COMLØ4 and COMLØ5.

2.6 COMLØ2.FOR

Authors:

Written for RT-11 by R. Kellogg, Measurement Concept Corp., 1979.
Modified for RSX-11M by J. Tims, Measurement Concept Corp., 1979.

COMLØ2 contains routines to support requests for

- Read/Write Image Memory
- Read/Write Graphic Memory

issued through CPIPSB (see Section 1.5).

2.7 COMLØ3.FOR

Authors:

Written for RT-11 by R. Kellogg, Measurement Concept Corp., 1979.
Modified for RSX-11M by J. Tims, Measurement Concept Corp., 1979.

COMLØ3 contains routines to support requests for

- Read/Write Pseudocolor Memory
- Read/Write Graphic Memory

Issued through CPIPSB (see Section 1.6).

2.8 COMLØ4.FOR

Authors:

Written for RT-11 by R. Kellogg, Measurement Concept Corp., 1979.
Modified for RSX-11M by J. Tims, Measurement Concept Corp., 1979.

COML04.FOR is not used by the RIPS system. It consists of routines to

- Read/Write COMTAL target cursor

and hence has no current function under RIPS (see Section 1.7).

2.9 COML05.FOR

Authors:

Written for RT-11 by R. Kellogg, Measurement Concept Corp., 1979.
Modified for RSX-11M by J. Tims, Measurement Concept Corp., 1979.

COML05 contains routines to support requests for

- Xmit block of text to COMTAL

This function is used in RIPS to format commands to the COMTAL to be interpreted by its MACRO interpreter.

Although this is the only RIPS function supported, COML05 contains many other capabilities not used by RIPS (see Section 1.8).

- Read block of text from COMTAL
- Xmit data block to COMTAL
- Read data block from COMTAL
- Write block of PDP 11/40 memory to COMTAL (to be used when additional LSI-11 memory is purchased for down-line loading of COMTAL programs)
- Read block of COMTAL LSI-11 memory from the PDP-11/40.

2.10 IWRITW.MAC

Author:

J. Tims, Measurement Concept Corp., 1979.

IWRITW was written to mimic the RT-11 I/O function IWRITW, which was used by COMPIP. It also includes code to mimic the RT-11 function routine IADDR, which would return the starting address of a FORTRAN parameter. No such routine exists within RSX-11M FORTRAN.

IWRITW accepts I/O requests that "look like" RT-11 requests and converts them into RSX-11M QIOW$ requests.

Also included in IWRITW.MAC is subroutine IBYTEX, which is used to swap high-and-low order bytes within a PDP-11 word.

2.11 Building the Modified RIPS Task

To construct a new executable task image after modifying one or more of the source modules described previously, it is necessary to perform only three steps:

a. MCR>SET /UIC = [11, 11]
b. Recompile or reassemble the affected source modules to obtain corresponding object modules.
c. MCR>TKB @DK3:RIPSLNK

It is assumed that:

- RSX-11M system disk is mounted on drive Ø
- MCC:RSX is mounted on drive 3.
- PAR Development Disk is mounted on drive 2.

This will build the task "RIPS.TSK" on DK3 :[11, 11].

2.12 To Run "RIPS" with the COMTAL

```
MCR>LOA ZB:          (Load the COMTAL Device Driver)
MCR>INS DK3:RIPS     (Install the task)
MCR>RUN RIPS         (Run the task)
```

Once executing, RIPS prompts are self-explanatory.

2.13 Integration of Mc^2 Software With Other User's Programs

Other users may gain access to the COMTAL under RSX-11M by including certain of the Mc^2 object modules in their task builder (TKB) command lines. Normally, the modules required will be:

COMLØ1.OBJ
COMLØ2.OBJ
COMLØ3.OBJ
COMLØ4.OBJ
COMLØ5.OBJ
IWRITW.OBJ

This will give the user access to all the I/O functions described in the documentation of COMLØ2 - COMLØ5. If it is determined that all of these functions are not necessary for the user's program, it may be possible to remove one of the modules COMLØ2 - COMLØ5 if none of the functions within the module are required by the user. In any case, COMLØ1 and IWRITW are required, and cannot be omitted.

```
        .ROOT DK2:EXEC-ERRFLG-SIMACC-L1-%(L2,L3)
L1:     .FCTR DK2:SETFDB-DK2:EERR-DK2:GETNUM-DK2:FSMUT-DK2:FCLOSE-LA1
LA1:    .FCTR DK2:FCRE-DK2:FCREO-DK2:FRET-DK2:FRETO-DK2:FILE1-LB1
LB1:    .FCTR DK2:BLDISP-DK2:GETEXT-DK2:FDEL-DK2:DSKXFR-LC1
LC1:    .FCTR DK2:FILEIO-DK2:TTI-DK2:TTO-LD1
LD1:    .FCTR DK2:TELEIO-LE1
LE1:    .FCTR DK2:TTYGRF-LF1
LF1:    .FCTR DK2:PLOT-LG1
LG1:    .FCTR DK2:SAVER-LH1
LH1:    .FCTR DK2:ADEXTN-DK2:GTEXTN-DK2:DUPCHK-DK2:LOGINP-LI1
LI1:    .FCTR DK2:SETLUN-LJ1
LJ1:    .FCTR DK2:TTYIO-DK2:CB2DA-DK2:UCLOSE-LK1
LK1:    .FCTR DK2:FLPTPR-DK2:FPPMOV-DK2:LDC-LL1
LL1:    .FCTR DK2:MODD02-DK2:MODF02-DK2:SARFPS-LM1
LM1:    .FCTR DK2:SETCCX-DK2:STC-DK2:DBLADD-LN1
LN1:    .FCTR DK2:DBLDIV-DK2:DBLMUL-DK2:DBLSUB-LO1
LO1:    .FCTR DK2:FLT-DK2:FLTD-DK2:FLTL-LP1
LP1:    .FCTR DK2:FLTLD-DK2:IDX-DK2:IFX-LQ1
LQ1:    .FCTR DK2:LDX-DK2:LFX-DK2:MODD13-LR1
LR1:    .FCTR DK2:MODF13-DK2:STRDBL-DK2:STSNGL-LS1
LS1:    .FCTR DK2:OPTION-DK2:REMAIN
L2:     .FCTR DK2:FSYSIN-DK2:SETSIZ-DK2:SETDEF
L3:     .FCTR DK2:CALSUB-(Z1,Z2)
Z1:     .FCTR DK2:FPTR-DK2:FDPTR-(L4,L5,L6,L7,L8,L9,L10,L11,L12,L14,L15,L33)
Z2:     .FCTR DK2:FPTR2-DK2:FDPTR2-(L13,L27)
L4:     .FCTR DK2:TEKDSP
L5:     .FCTR DK2:TAPINP-DK2:TAPSUB
L6:     .FCTR DK2:TAPOUT-DK2:TPREW-DK2:TPSKIP-DK2:TAPSUB
L7:     .FCTR DK2:SIMSMO-DK2:SMOOTH
L8:     .FCTR DK2:PURGE-DK2:LOGERR-DK2:FRNM-DK2:RENAME
L9:     .FCTR DK2:ELCHNG-DK2:TRNFTN
L10:    .FCTR DK2:USRPRG-DK2:FDPTR2-DK2:FPTR2
L11:    .FCTR DK2:PTEDGE
L12:    .FCTR DK2:DENHST-DK2:HISPLT-DK2:FPASC-DK2:FINDPT
L13:    .FCTR DK2:LINES-DK2:SMOTH2
L14:    .FCTR DK2:NORMLZ
L15:    .FCTR   (L16,L17,L18,L19,L20,L21,L22,L23,L24,L25,L26,L28,L29,L30,L31,L32)
L16:    .FCTR   DK2:UTDSMO
L17:    .FCTR   DK2:PARCOR-DK2:OUTCAR-DK2:TRNFTN-(L17A,L17B,L17C,L17D,L17E)
L17A:   .FCTR   DK2:ARSLCT
L17B:   .FCTR   DK2:RTIOMG
L17C:   .FCTR   DK2:COMBIM
L17D:   .FCTR   DK2:DIFFAC
L17E:   .FCTR   DK2:PULINR
L18:    .FCTR   DK2:SPBIAS
L19:    .FCTR   DK2:SHIFTI
L20:    .FCTR   DK2:CKTEXT
L21:    .FCTR   DK2:CTFILE
L22:    .FCTR   DK2:GTFILE
L23:    .FCTR   DK2:ODDLD
L24:    .FCTR   DK2:DECIM
L25:    .FCTR   DK2:MODREP
L26:    .FCTR   DK2:SFILL
L27:    .FCTR   DK2:CLCV
L28:    .FCTR   DK2:DIRCTV
L29:    .FCTP   DK2:KEYARA
L30:    .FCTR   DK2:LPHDCP-DK2:LNPRNT-DK2:DUMP
```

```
L31:    .FCTR   DK2:ADFILL
L32:    .FCTR   DK2:RGSTAT-DK2:SPAME2-DK2:FINDPT-DK2:SROOT-DK2:LNPRNT
L33:    .FCTR   DK2:CPIPSB-DK2:IURITU-L34
L34:    .FCTR   L35-L36-L37-L38-L39
L35:    .FCTR   DK2:COML01
L36:    .FCTR   DK2:COML02
L37:    .FCTR   DK2:COML03
L38:    .FCTR   DK2:COML04
L39:    .FCTR   DK2:COML05
        .PSECT  CRSIZ,RW,D,GBL,REL,OVR
        .PSECT  CRSIZ2,RW,D,GBL,REL,OVR
        .PSECT  DIGCNT,RW,D,GBL,REL,OVR
        .PSECT  ERRFLG,RW,D,GBL,REL,OVR
        .PSECT  FRECOR,RW,D,GBL,REL,OVR
        .PSECT  FRMTBL,RW,D,GBL,REL,OVR
        .PSECT  FSLOTS,RW,D,GBL,REL,OVR
        .PSECT  LINEC,RW,D,GBL,REL,OVR
        .PSECT  LOGFLG,RW,D,GBL,REL,OVR
        .PSECT  NAMTAB,RW,D,GBL,REL,OVR
        .PSECT  SIMACC,RW,D,GBL,REL,OVR
        .PSECT  TTLUN,RW,D,GBL,REL,OVR
        .PSECT  TTYADR,RW,D,GBL,REL,OVR
        .PSECT  TTYBUF,RW,D,GBL,REL,OVR
        .PSECT  TTYLUN,RW,D,GBL,REL,OVR
        .PSECT  DEVUIC,RW,D,GBL,REL,OVR
        .PSECT  INNAM,RW,D,GBL,REL,OVR
        .PSECT  OUTNAM,RW,D,GBL,REL,OVR
        .PSECT  COMTAL,RW,D,GBL,REL,OVR
        .PSECT  TC,RW,D,GBL,REL,OVR
        .PSECT  DDB,RW,D,GBL,REL,OVR
        .PSECT  DATAA,RW,D,GBL,REL,OVR
        .PSECT  COMVAL,RW,D,GBL,REL,OVR
        .PSECT  LINE,RW,D,GBL,REL,OVR
        .END
PIP>
```

```
C
C     PROGRAM NAME: CALSUB                          TYPE: SUBROUTINE
C
C     PROGRAMMER: ALBERT J. FRANKLIN                DATE OF LAST UPDATE: 15-JAN-79
C
C         PROGRAM FUNCTION:
C     CALSUB CALLS THE REQUESTED OPTION SUBROUTINE.
C
C         ENTRY POINTS AND CALLING SEQUENCE:
C     CALL CALSUB (IF,IS,CORE,ISIZE)
C     INPUTS: IF,IS,CORE,ISIZE
C     OUTPUTS: NONE
C
C         DETAILED PROGRAM DESCRIPTION:
C     CALSUB GOES TO THE OPTION LIST FOR FRAME IF, AND CHECKS FOR OPTION
C     NUMBER IS AMONG THE LIST OF IMPLEMENTED OPTIONS.  IF THE OPTION IS
C     MATCHED, THE APPROPRIATE SUBROUTINE IS CALLED; IF NOT, AN ERROR
C     MESSAGE IS PRINTED AND NO FURTHER ACTION IS TAKEN.
C
C         SUBROUTINES CALLED (EXTERNAL):
C     ABSDIF - ABSOLUTE DIFFERENCE
C     ABSUS      - UNTD. ABS. VALUE SMOOTH
C     ADFILL - ADAPTIVE FILL
C     ARSLCT - RECTANGULAR AREA SELECT
C     AVE - AVERAGE IMAGES
C     BOXSMO     - BOX SMOOTH
C     CKTEXT - EXAMINE HEADER TEXT
C     CLCU - CLOSED CURVE
C     CTFILE - CREATE TEST IMAGE
C     DECIM - DECIMATION
C     DIFFAC - SCALED WEIGHTED COMBINATION
C     DIRCTY - DIRECTORY LISTING UTILITY
C     EDGPRN - LIST BINARY IMAGE
C     ELCHNG - THRESHOLD AND ELEMENT CHANGE
C     FDEL - DELETE A DISK FILE
C     FRNM - RENAME A DISK FILE
C     GRYDMP - LIST IMAGE GREY VALUES
C     GTFILE - GENERATE TEST IMAGE
C     IMAGE - IMAGE HISTOGRAMS
C     KEYARA - SPECIFY REGION COORDINATES
C     LINES      - AREA EDGE DETECTION
C     LOGERR - CONTROL THE FOUR LOG OPTIONS
C     MAED       - AREA EDGE DETECTION MAX.
C     MODREP - MODAL REPLACEMENT
C     NDOT - ODD DOT ELIMINATION
C     NORMLZ - NORMALIZE AND RANGE CHANGE
C     ODDLIN - ODD LINE ELIMINATION
```

```
C     PTEDGE - POINT EDGE DETECTION
C     PULINR - ARBITRARY FUNCTION
C     REGION - REGION HISTOGRAMS
C     RGSTAT      - LIST REGION STATISTICS
C     RNCHNG      - RANGE CHANGE
C     RTIOMG - RATIO IMAGES
C     SCALE - SCALED DIFFERENCE
```



```
C     SFILL - SIMPLE FILL
C     SHIFTI - SHIFT IMAGE
C     SIMSMO      - M X N SMOOTH
C     SMGLH       - IMAGE ROW INTENSITY PROFILE
C     SPBIAS - ADD CONSTANT
C     TAPINP - INPUT AN IMAGE FILE FROM TAPE
C     TAPOUT - OUTPUT AN IMAGE FILE TO TAPE
C     TEKDSP - DISPLAY A BINARY IMAGE ON THE TEKTRONIX 4014
C     THRSH       - THRESHOLD IMAGE
C     TPREW - REWIND MAGTAPE ON UNIT 0
C     TPSKIP - SKIP FILES ON MAPTAPE MOUNTED ON UNIT 0
C     TTYOUT - OUTPUT CHARACTER STRING TO TTY
C     USRPRG - ANY USER PROGRAM
C     WTSMO - WEIGHTED SMOOTH AND ABSOLUTE VALUE WEIGHTED SMOOTH
C
C         INTERNAL VARIABLES AND ARRAYS (FORTRAN TYPE CONVENTION):
C     IER - ERROR STATUS
C
C         GLOBAL VARIABLES AND ARRAYS (FORTRAN TYPE CONVENTION):
C     CORE - BYTE ARRAY (ISIZE) USER DATA BUFFER AREA
C     IF - FRAME NUMBER
C     IS - OPTION NUMBER
C     ISIZE - SIZE OF USER DATA BUFFER AREA IN BYTES
C     LCOUNT - TEKTRONIX LINE COUNT
C
C         SPECIAL COMMENTS:
C     THE FREE CORE AREA AND ITS SIZE IN BYTES ARE PASSED THROUGH
C     CALSUB TO THE CALL ROUTINES BY PARAMETERS RATHER THAN IN
C     COMMON.  THIS IS SO ONLY EXEC MUST BE CHANGED WHEN THE SIZE
C     OF FREE CORE CHANGES.
C
0001        SUBROUTINE CALSUB (IF,IS,CORE,ISIZE)
C
0002        BYTE CORE(ISIZE)
C
C-------------------------------------------------------------------------
0003        COMMON /LINEC/LCOUNT
C-------------------------------------------------------------------------
C
C     ////////// BEGIN EXECUTABLE INSTRUCTIONS //////////
```

```
C
C     GO TO CODE FOR FRAME IF.
C
0004      GO TO (21,1,2,3,4,5,6,7,21,21,21,11,12,13,14,
     1 21,21,21,21,21,21), IF + 1
C
C     CODE FOR FRAME 1.
C
0005  1   IF (IS .EQ. 1) CALL ARSLCT
0007      IF (IS .EQ. 2) CALL KEYARA
0009      IF (IS .EQ. 3) CALL SHIFTI
0011      IF (IS .EQ. 4) CALL DECIM
0013      IF (IS .GT. 4) GO TO 21
0015      RETURN
```



```
C     CODE FOR FRAME 2.
C
0016  2   IF (IS .EQ. 1) CALL SPBIAS
0018      IF (IS .EQ. 2) CALL THRSH
0020      IF (IS .EQ. 3) CALL NORMLZ
0022      IF (IS .EQ. 4) CALL ELCHNG
0024      IF (IS .EQ. 5) CALL RMCHNG
0026      IF (IS .EQ. 6) CALL PULINR
0028      IF (IS .GT. 6) GO TO 21
0030      RETURN
C
C     CODE FOR FRAME 3
C
0031  3   IF (IS .EQ. 1) CALL IMAGE
0033      IF (IS .EQ. 2) CALL REGION
0035      IF (IS .EQ. 3) CALL RGSTAT
0037      IF (IS .EQ. 5) CALL SNGLH
0039      IF (IS .EQ. 4 .OR. IS .GT. 5) GO TO 21
0041      RETURN
C
C     CODE FOR FRAME 4
C
0042  4   IF (IS .EQ. 1) CALL BOXSMO
0044      IF (IS .EQ. 2) CALL SIMSMO
0046      IF (IS .EQ. 3) CALL UTSMO
0048      IF (IS .EQ. 4) CALL ABSUS
0050      IF (IS .GT. 4) GO TO 21
0052      RETURN
C
C     CODE FOR FRAME 5
C
0053  5   IF (IS .EQ. 1) CALL AVE
0055      IF (IS .EQ. 2) CALL SCALE
```

```
0057          IF (IS .EQ. 3) CALL ABSDIF
0059          IF (IS .EQ. 4) CALL DIFFAC
0061          IF (IS .EQ. 5) CALL RTIOMG
0063          IF (IS .GT. 5) GO TO 21
0065          RETURN
        C
        C     CODE FOR FRAME 6
        C
0066    6     IF (IS .EQ. 1) CALL MODREP
0068          IF (IS .EQ. 2) CALL MDOT
0070          IF (IS .EQ. 3) CALL ODDLIN
0072          IF (IS .GT. 3) GO TO 21
0074          RETURN
        C     CODE FOR FRAME 7
        C
0075    7     IF (IS .EQ. 1)CALL PTEDGE
0077          IF (IS .EQ. 2)CALL LINES
0079          IF (IS .EQ. 3) CALL MAED
0081          IF (IS .EQ. 4) CALL SFILL
0083          IF (IS .EQ. 5) CALL ADFILL
0085          IF (IS .EQ. 6) CALL CLCV
```



```
0087          IF (IS .EQ. 7) CALL EDGPRN
0089          IF(IS .GT. 7)GO TO 21
0091          RETURN
        C
        C     CODE FOR FRAME 11.
        C
0092    11    IF (IS .EQ. 1) CALL FDEL (-1,IER)
0094          IF (IS .EQ. 2) CALL FRNM (-1,IER)
0096          IF (IS .EQ. 3 .OR. IS .EQ. 4) CALL DIRCTY
0098          IF (IS .EQ. 5) CALL TAPINP (CORE,ISIZE,IER)
0100          IF (IS .EQ. 6) CALL TAPOUT (CORE,ISIZE,IER)
0102          IF (IS .EQ. 7) CALL TPSKIP (IER)
0104          IF (IS .EQ. 8) CALL TPREW (IER)
0106          IF (IS .LE. 8) RETURN
0108          GO TO 21
        C
        C     CODE FOR FRAME 12.
        C
0109    12    IF (IS .GT. 1) GO TO 21
0111          IF (IS .EQ. 1) CALL TEKDSP
0113          RETURN
        C
        C
        C     CODE FOR FRAME 13
        C
0114    13    IF (IS .EQ. 2) CALL LOGERR (IER)
```

```
0116          IF(IS .EQ. 3)CALL USRPRG
0118          IF (IS .EQ. 4) CALL GTFILE
0120          IF (IS .EQ. 5) CALL CTFILE
0122          IF (IS .EQ. 6) CALL CKTEXT
0124          IF (IS .EQ. 7) CALL GRYDMP
0126          IF (IS .EQ. 1 .OR. IS .GT. 7) GO TO 21
0128          RETURN
              C     CODE FOR FRAME 14
0129    14    IF (IS .EQ. 1) CALL IWIMF
0131          IF (IS .EQ. 2) CALL IRIMF
0133          IF (IS .EQ. 3) CALL IWGMF
0135          IF (IS .EQ. 4) CALL IRGMF
0137          IF (IS .EQ. 5) CALL IWFMF
0139          IF (IS .EQ. 6) CALL IRFMF
0141          IF (IS .EQ. 7) CALL IWPSF
0143          IF (IS .EQ. 8) CALL IRPSF
0145          IF (IS .EQ. 9) CALL IWCBF
0147          IF (IS .EQ. 0 .OR. IS .GT. 9) GO TO 21
0149          RETURN
              C     PRINT MESSAGE SAYING SUBROUTINE NOT OPERATIVE.
              C
0150    21    CALL TTYOUT ('NOT OPERATIVE!!')
0151          LCOUNT = LCOUNT + 1
0152          RETURN
0153          END
```



```
FORTRAN IV      STORAGE MAP

NAME    OFFSET  ATTRIBUTES

CORE    000020  LOGICAL*1 PARAMETER ARRAY (ISIZE)
ISIZE   000022  INTEGER*2 PARAMETER VARIABLE
IF      000014  INTEGER*2 PARAMETER VARIABLE
IS      000016  INTEGER*2 PARAMETER VARIABLE
ARSLCT  000000  REAL*4    PROCEDURE
KEYAPA  000000  INTEGER*2 PROCEDURE
SHIFTI  000000  REAL*4    PROCEDURE
DECIM   000000  REAL*4    PROCEDURE
SPBIAS  000000  REAL*4    PROCEDURE
THRSH   000000  REAL*4    PROCEDURE
NORMLZ  000000  INTEGER*2 PROCEDURE
ELCHNG  000000  REAL*4    PROCEDURE
RNCHNG  000000  REAL*4    PROCEDURE
PWLINR  000000  REAL*4    PROCEDURE
IMAGE   000000  INTEGER*2 PROCEDURE
REGION  000000  REAL*4    PROCEDURE
RGSTAT  000000  REAL*4    PROCEDURE
SNGLM   000000  REAL*4    PROCEDURE
BOXSMO  000000  REAL*4    PROCEDURE
SIMSMO  000000  REAL*4    PROCEDURE
UTSMO   000000  REAL*4    PROCEDURE
```

```
AESWS   000000  REAL*4     PROCEDURE
AVE     000000  REAL*4     PROCEDURE
SCALE   000000  REAL*4     PROCEDURE
AESDIF  000000  REAL*4     PROCEDURE
DIFFAC  000000  REAL*4     PROCEDURE
RTIOMG  000000  REAL*4     PROCEDURE
MODPEP  000000  INTEGER*2  PROCEDURE
NDOT    000000  INTEGER*2  PROCEDURE
ODDLIN  000000  REAL*4     PROCEDURE
PTEDGE  000000  REAL*4     PROCEDURE
LINES   000000  INTEGER*2  PROCEDURE
MAED    000000  INTEGER*2  PROCEDURE
SFILL   000000  REAL*4     PROCEDURE
ADFILL  000000  REAL*4     PROCEDURE
CLCV    000000  REAL*4     PROCEDURE
EDGPRN  000000  REAL*4     PROCEDURE
FDEL    000000  REAL*4     PROCEDURE
IER     000044  INTEGER*2  VARIABLE
FRNM    000000  REAL*4     PROCEDURE
DIRCTY  000000  REAL*4     PROCEDURE
TAPINP  000000  REAL*4     PROCEDURE
TAPOUT  000000  REAL*4     PROCEDURE
TPSKIP  000000  REAL*4     PROCEDURE
TPREW   000000  REAL*4     PROCEDURE
TEKDSP  000000  REAL*4     PROCEDURE
LOGERR  000000  INTEGER*2  PROCEDURE
USRPRG  000000  REAL*4     PROCEDURE
GTFILE  000000  REAL*4     PROCEDURE
CTFILE  000000  REAL*4     PROCEDURE
CKTEXT  000000  REAL*4     PROCEDURE
GRYDMP  000000  REAL*4     PROCEDURE
```



```
FORTRAN IV      STORAGE MAP

NAME    OFFSET  ATTRIBUTES

IWIMF   000000  INTEGER*2 PROCEDURE
IRIMF   000000  INTEGER*2 PROCEDURE
IWGMF   000000  INTEGER*2 PROCEDURE
IRGMF   000000  INTEGER*2 PROCEDURE
IWFMF   000000  INTEGER*2 PROCEDURE
IRFMF   000000  INTEGER*2 PROCEDURE
IWPSF   000000  INTEGER*2 PROCEDURE
IRPSF   000000  INTEGER*2 PROCEDURE
IWCBF   000000  INTEGER*2 PROCEDURE
TTYOUT  000000  REAL*4    PROCEDURE

COMMON BLOCK /LINEC/    LENGTH 000002

LCOUNT  000000  INTEGER*2 VARIABLE
```

```
FORTRAN IV      V01C-03         MON 29-OCT-79 14:08:35          PAGE 001
CORE*OSK, UIC*[1,123]                                ,TT0:*DK3:CPIPSB

C       ,                       FILE NAME: CPIPSB.FTN
C       ,                       DATE LAST EDITED: 21-JUN-79
C
C                                               10-JUL-79
C       ,                       HISTORY:
C       ,                       DATE:           MODIFICATION
C       ,                       -----           ------------
C       ,                       19-MAR-79       INITAL WRITING
C       ,                       21-MAR-79       ADD MORE SUBS
C       ,                       18-APR-79       ADD CLEAR SCREEN
C       ,                                       TO 'IWIMF', AND
C       ,                                       ADD ' IUCBF'
C                               21-JUN-79    CONVERSION TO RIPS
C
C       ,   THIS FILE CONTAINS THE MAIN SUBROUTINES FOR THE
C       ,   RIPS COMTAL MANIPULATIONS AND CALLS SEVERAL OF THE COMTAL LIBRARY
C       ,   ROUTINES AS WELL AS OTHER RIPS FILE READING ROUTINES.
C
C       ,   THE MAIN ROUTINES ARE:
C       ,           IWIMF  -  READ A RIPS FILE DISPLAY ON COMTAL
C       ,           IRIMF  -  WRITE A RIPS FILE FROM THE COMTAL
C       ,           IWGMF  -  READ A RIPS GRAPHICS FILE TO COMTAL
C       ,           IRGMF  -  WRITE A RIPS GRAPHICS FILE FROM COMTAL
C       ,           IWFMF  -  READ A RIPS FUNCTION FILE TO COMTAL
C       ,           IRFMF  -  WRITE A RIPS FUNCTION FILE FROM COMTAL
C       ,           IWPSF  -  READ A RIPS PSEUDO-COLOR FILE TO COMTAL
C       ,           IRPSF  -  WRITE A RIPS PSEUDO-COLOR FILE FROM COMTAL
C       ,           IWCBF  -  WRITE COMMAND BLOCK TO COMTAL
C
0001        SUBROUTINE IWIMF
C
C       ,   FUNCTIONAL DESCRIPTION:
C
C       ,   THIS ROUTINE WILL READ A RIPS ' IMAGE ' TYPE FILE AND
C       ,   WRITE IT TO ONE OF THE IMAGE MEMORIES IN THE COMTAL.
C
C       ,   1) QUERY USER FOR IMAGE NUMBER
```

```
C     ,   3) OPEN FILE FOR READ/WRITE
C     ,   4) CLEAR SCREEN
C     ,   5) READ FILE, WRITE COMTAL
C
0002        LOGICAL*1 ITEXT(8),ONE,TWO,THREE,FOUR
0003        LOGICAL*1 NAME(32),IMG(4),INAM(50)
0004        INTEGER NFTYP(2),FREECR(4096)
0005        INTEGER BUF(256)
```



```
0006        COMMON      /FRECOR/ FREECR,/IMAQUE/IMAQUE
C
0007        COMMON      /INNAM/INAM,/NAMTAB/NAME,/COMTAL/BUF
0008        DATA IMG /'.','I','M','G'/
0009        DATA ITEXT /'C','L',' ',' ','I',' ',' ','1',' ',' ','8'/
0010        DATA ONE /'1'/, TWO /'2'/, THREE /'3'/, FOUR /'4'/
C     ,   1) QUERY USER
0011        ICNT=1
0012        CALL        SNGDEC(IMAQUE,ICNT,IMNO,IER)
0013        IF(IMNO.EQ.1) ITEXT(6) = ONE
0015        IF(IMNO.EQ.2) ITEXT(6) = TWO
0017        IF(IMNO.EQ.3) ITEXT(6) = THREE
0019        IF(IMNO.EQ.4) ITEXT(6) = FOUR
C
C
C     ,   READ RIPS FILE, WRITE TO COMTAL
C     ,   3) OPEN RIPS FILE
C
0021        CALL FRETO(INAM,NRECS,NRECSZ,NAME,IMG,LUN,BUF,IER)
C
C     ,   CALC OFFSET TO PUT IN CENTER OF SCREEN
C
0022        IXOFF = ( 128 - NRECSZ ) /2
0023        IF(IXOFF.LT.1) IXOFF = 1
C
0025        IYOFF = ( 256 - NRECS ) /2
0026        IF ( IYOFF .LT .1) IYOFF = 1
C
C     ,   CLEAR OUT BUFFER
C
0028        DO 900 J=1,256
C
0029        FREECR(J)= 0
C
0030  900   CONTINUE
C
C     ,   4) CLEAR IMAGE
0031        IER = IUCB(ITEXT)
C
C     ,   5) READ A RIPS RECORD , WRITE TO COMTAL
```

```
C
0032        DO 100 K=1,NRECS
C
0033        J = K -1 + IVOFF
C
C
C
0034        CALL FPTR(LUN,FREECR(IXOFF),256,K,1,0,LPTR,IEOF,IER)
0035        IER=IBYTEX(FREECR,256)
0036        IER=IWIM(IMMO,J,FREECR)
C
0037  100   CONTINUE
0038        CALL FCLOSE(LUN,IER)
C
```

```
FORTRAN IV      V01C-03         MON 29-OCT-79 14:08:35          PAGE 003
CORE=08K, UIC=[1,123]                                  ,TT0:=DK3:CPIPSB

0039        RETURN
0040        END
```



```
FORTRAN IV      STORAGE MAP

NAME    OFFSET  ATTRIBUTES

ITEXT   000014  LOGICAL*1 ARRAY (8)
IMG     000024  LOGICAL*1 ARRAY (4)
NETYP   000030  INTEGER*2 ARRAY (2)
ONE     000034  LOGICAL*1 VARIABLE
TWO     000035  LOGICAL*1 VARIABLE
THREE   000036  LOGICAL*1 VARIABLE
FOUR    000037  LOGICAL*1 VARIABLE
ICNT    000046  INTEGER*2 VARIABLE
SNGDEC  000000  REAL*4    PROCEDURE
IMMO    000050  INTEGER*2 VARIABLE
IER     000052  INTEGER*2 VARIABLE
FRETO   000000  REAL*4    PROCEDURE
NRECS   000054  INTEGER*2 VARIABLE
NRECSZ  000056  INTEGER*2 VARIABLE
LUN     000060  INTEGER*2 VARIABLE
IXOFF   000062  INTEGER*2 VARIABLE
IVOFF   000064  INTEGER*2 VARIABLE
J       000066  INTEGER*2 VARIABLE
IWCB    000000  INTEGER*2 PROCEDURE
```

```
K       000070  INTEGER*2 VARIABLE
FPTR    000000  REAL*4    PROCEDURE
LPTR    000072  INTEGER*2 VARIABLE
IEOF    000074  INTEGER*2 VARIABLE
IBYTE\  000000  INTEGER*2 PROCEDURE
IWIM    000000  INTEGER*2 PROCEDURE
FCLOSE  000000  REAL*4    PROCEDURE

COMMON BLOCK /FRECOR/   LENGTH 020000

FREECR  000000  INTEGER*2 ARRAY (4096)

COMMON BLOCK /IMAGUE/   LENGTH 000002

IMAGUE  000000  INTEGER*2 VARIABLE

COMMON BLOCK /INNAM/    LENGTH 000062

INAM    000000  LOGICAL*1 ARRAY (50)

COMMON BLOCK /NAMTAB/   LENGTH 000040

NAME    000000  LOGICAL*1 ARRAY (32)

COMMON BLOCK /COMTAL/   LENGTH 001000

BUF     000000  INTEGER*2 ARRAY (256)
```



```
FORTRAN IV      V01C-03         MON 29-OCT-79 14:09:10          PAGE 001
CORE=08K, UIC=[1,123]                                  ,TT0:=DK3:CPIPSB
      C
0001        SUBROUTINE IRIMF
      C
      C     ,   FUNCTIONAL DESCRIPTION:
      C
      C     ,   THIS ROUTINE WILL READ A COMTAL IMAGE AND
      C     ,   WRITE IT TO A RIPS ' IMAGE '  FILE.
      C
      C
      C     ,   1) QUERY USER FOR IMAGE NUMBER
      C     ,   2) OPEN FILE FOR READ/WRITE
      C     ,   3) READ COMTAL, WRITE FILE
      C
0002        INTEGER NAFTYP(2),FREECR(4096)
0003        LOGICAL*1 NAME(32)
0004        INTEGER BUF(256)
      C
0005        COMMON /OUTNAM/ONAM,/COMTAL/BUF,/FRECOR/FREECR
0006        COMMON /NAMTAB/NAME,/IMAGUE/IMAGUE
```

```
0007            DATA   NAFTYP /'.I','MG'/
                C
                C       1) QUERY USER
0008            ICNT=1
0009            CALL        SNGDEC(IMAGUE,ICNT,IMNO,IER)
                C       ,   READ CONTAL ,WRITE RIPS FILE
                C
                C
                C       CREATE THE RIPS FILE BEFORE WRITING
0010            CALL FCREO(ONAM,256,256,NAME,NAFTYP,LUN,BUF,IER)
                C
0011            DO 200      J=1,256
0012            IER=IRIM(IMNO,J,FREECR)
0013            IER=IBYTEX(FREECR,256)
                C       READ THE IMAGE FROM THE CONTAL
                C
0014            CALL        FPTR(LUN,FREECR,256,J,1,1,LPTR,IEOF,IER)
                C       NOW WRITE IT TO THE FILE
0015  200       CONTINUE
                C
0016            CALL FCLOSE(LUN,IER)
                C
0017            RETURN
0018            END
```



```
FORTRAN IV      STORAGE MAP

NAME    OFFSET  ATTRIBUTES

NAFTYP  000014  INTEGER*2 ARRAY (2)
ICNT    000024  INTEGER*2 VARIABLE
SNGDEC  000000  REAL*4    PROCEDURE
IMNO    000026  INTEGER*2 VARIABLE
IER     000030  INTEGER*2 VARIABLE
FCREO   000000  REAL*4    PROCEDURE
LUN     000032  INTEGER*2 VARIABLE
J       000034  INTEGER*2 VARIABLE
IRIM    000000  INTEGER*2 PROCEDURE
IBYTEX  000000  INTEGER*2 PROCEDURE
FPTR    000000  REAL*4    PROCEDURE
LPTR    000036  INTEGER*2 VARIABLE
IEOF    000040  INTEGER*2 VARIABLE
FCLOSE  000000  REAL*4    PROCEDURE

COMMON BLOCK /OUTNAM/   LENGTH 000004

ONAM    000000  REAL*4    VARIABLE

COMMON BLOCK /CONTAL/   LENGTH 001000
```

```
BUF     000000  INTEGER*2 ARRAY (256)
COMMON BLOCK /FRECOR/   LENGTH 020000
FREECR  000000  INTEGER*2 ARRAY (4096)
COMMON BLOCK /NAMTAB/   LENGTH 000040
NAME    000000  LOGICAL*1 ARRAY (32)
COMMON BLOCK /IMAGUE/   LENGTH 000002
IMAGUE  000000  INTEGER*2 VARIABLE
```

```
FORTRAN IV      V01C-03         MON 29-OCT-79 14:09:31          PAGE 001
CORE=08K, UIC=[1,123]                                   ,TT0:=DK3:CPIPSB
```

```
        C
0001        SUBROUTINE IUPSF
        C
        CCCCCCCCCCC
        C       ,   FUNCTIONAL DESCRIPTION:
        C
        C       ,   THIS ROUTINE WILL READ A RIPS PSEUDO-COLOR FILE
        C       ,   AND WRITE IT TO THE COMTAL PSEUDO-COLOR MEMORY.
        C
        C
        C       ,   1)
        C       ,   2) OPEN FILE FOR READ/WRITE
        C       ,   3) READ FILE, WRITE COMTAL
        C       ,   4) , CLOSE RIPS FILE
        CCC
        C
0002        INTEGER BUF(256),PSE(2),FREECR(4096)
0003        LOGICAL*1  NAME(32)
0004        COMMON /INNAM/INAM,/FRECOR/FREECR,/COMTAL/BUF
0005        COMMON /NAMTAB/NAME
0006        DATA PSE /'.P','SE'/
        C
        CCC
        C
0007        CALL FRETO(INAM,NRECS,NRECS2,NAME,PSE,LUN,BUF,IER)
        C
        CC      ,   4) READ RIPS FILE, WRITE TO COMTAL PSEUDO-COLOR
0008        CALL FPTR(LUN,FREECR,512,1,2,0,LPTR,IEOF,IER)
        C
0009        IER = IUPS (FREECR)
        C
        CCC     ,   5) CLOSE RIPS FILE, CLOSE COMTAL
        C
0010        CALL        FCLOSE(LUN,IER)
0011        RETURN
0012        END
```

```
FORTRAN IV      STORAGE MAP

NAME    OFFSET  ATTRIBUTES

PSE     000014  INTEGER*2 ARRAY (2)
FRETO   000000  REAL*4    PROCEDURE
NRECS   000030  INTEGER*2 VARIABLE
NRECSZ  000032  INTEGER*2 VARIABLE
LUN     000034  INTEGER*2 VARIABLE
IER     000036  INTEGER*2 VARIABLE
FPTR    000000  REAL*4    PROCEDURE
LPTR    000040  INTEGER*2 VARIABLE
IEOF    000042  INTEGER*2 VARIABLE
IWPS    000000  INTEGER*2 PROCEDURE
FCLOSE  000000  REAL*4    PROCEDURE

COMMON BLOCK /INNAM/    LENGTH 000002

INAM    000000  INTEGER*2 VARIABLE

COMMON BLOCK /FRECOR/   LENGTH 020000

FREECR  000000  INTEGER*2 ARRAY (4096)

COMMON BLOCK /COMTAL/   LENGTH 001000

BUF     000000  INTEGER*2 ARRAY (256)

COMMON BLOCK /NAMTAB/   LENGTH 000040

NAME    000000  LOGICAL*1 ARRAY (32)
```



```
FORTRAN IV      V01C-03         MON 29-OCT-79 14:09:42          PAGE 001
CORE=08K, UIC=[1,123]                                  ,TT0:=DK3:CPIPSB

      C
0001        SUBROUTINE IRPSF
      C
      C     ,   FUNCTIONAL DESCRIPTION:
      C
      C     ,   THIS ROUTINE WILL READ THE COMTAL PSEUDO-COLOR MEMORY
      C     ,   AND WRITE IT TO A RIPS PSEUDO-COLOR FILE.
```

```
C
C
C     1)
C     2) OPEN FILE FOR READ/WRITE
C     3) READ COMTAL, WRITE FILE
C     5) , CLOSE RIPS FILE
C
      INTEGER NAFTYP(2),FREECR(4096),BUF(256)
      LOGICAL*1 NAME(32)
      COMMON /FRECOR/ FREECR
      COMMON /OUTNAM/ONAM,/COMTAL/BUF,/NAMTAB/ NAME
      DATA    NAFTYP /'.P','SE'/
     C,   1)
C
      NRECS = 1
      NRECSZ= 256
      CALL FCREO(ONAM,NRECSZ,2,NAME,NAFTYP,LUN,BUF,IER)
C
C     ,   4) READ COMTAL, WRITE TO RIPS PSEUDO-COLOR
C
      IER = IRPS (FREECR)
      CALL FPTR(LUN,FREECR,NRECSZ,1,2,1,LPTR,IEOF,IER)
C
C     ,   5) CLOSE RIPS FILE, CLOSE COMTAL
C
      CALL FCLOSE(LUN,IER)
C
      RETURN
      END
```



```
FORTRAN IV      STORAGE MAP

NAME    OFFSET  ATTRIBUTES

NAFTYP  000014  INTEGER*2 ARRAY (2)
NRECS   000024  INTEGER*2 VARIABLE
NRECSZ  000026  INTEGER*2 VARIABLE
FCREO   000000  REAL*4    PROCEDURE
LUN     000030  INTEGER*2 VARIABLE
IER     000032  INTEGER*2 VARIABLE
IRPS    000000  INTEGER*2 PROCEDURE
FPTR    000000  REAL*4    PROCEDURE
LPTR    000034  INTEGER*2 VARIABLE
IEOF    000036  INTEGER*2 VARIABLE
FCLOSE  000000  REAL*4    PROCEDURE

COMMON BLOCK /FRECOR/   LENGTH 020000

FREECR  000000  INTEGER*2 ARRAY (4096)
```

```
COMMON BLOCK  OUTNAM/   LENGTH 000004
ONAM     000000  REAL*4    VARIABLE
COMMON BLOCK /COMTAL/   LENGTH 001000
BUF      000000  INTEGER*2 ARRAY (256)
COMMON BLOCK /NAMTAB/   LENGTH 000040
NAME     000000  LOGICAL*1 ARRAY (32)



FORTRAN IV      V01C-03         MON 29-OCT-79 14:10:04          PAGE 001
CORE=08K, UIC=[1,123]                                  ,TT0:=DK3:CPIPSB

      C
0001          SUBROUTINE IUGMF
      C
      C       ,   FUNCTIONAL DESCRIPTION:
      C
      C       ,   THIS ROUTINE WILL READ A RIPS GRAPHICS (.GRA) FILE
      C       ,   AND WRITE IT TO THE COMTAL GRAPHICS MEMORY.
      C
      C
      C       ,   1)
      C       ,   2) OPEN FILE FOR READ/WRITE
      C       ,   3) READ FILE, WRITE COMTAL
      C       ,   5) , CLOSE RIPS FILE
      C
0002          INTEGER BUF(256),FREECR(4096),GRA(2)
0003          LOGICAL*1 NAME(32)
0004          COMMON /FRECOR/ FREECR,/COMTAL/BUF,/NAMTAB/NAME
0005          COMMON /INNAM/INAM,/IGRAQU/IGRAQU
0006          DATA GRA /'.G','RA'/
      C
      C
      C       ,   1) QUERY USER FOR GRAPHICS NUMBER
      C
0007          ICNT=1
0008          CALL          SNGDEC(IGRAQU,ICNT,INNO,IER)
      C       ,   2) OPEN RIPS FILE
0009          CALL FRETO(INAM,NRECS,NRECSZ,NAME,GRA,LUN,BUF,IER)
      C       ,   4) READ RIPS FILE, WRITE TO COMTAL GRAPHICS MEMORY
      C
0010          DO 100 J=1,256
      C
0011          CALL FPTR(LUN,FREECR,256,J,1,0,LPTR,IEOF,IER)
0012          IER = IUGM (INNO,J,FREECR)
      C
0013    100   CONTINUE
```

```
C
C     ,   5) CLOSE RIPS FILE, CLOSE COMTAL
C
0014        CALL FCLOSE(LUN,IER)
0015        RETURN
0016        END
```

```
FORTRAN IV      STORAGE MAP

NAME    OFFSET  ATTRIBUTES

GRA     000014  INTEGER*2 ARRAY (2)
ICNT    000026  INTEGER*2 VARIABLE
SNGDEC  000000  REAL*4    PROCEDURE
IMNO    000030  INTEGER*2 VARIABLE
IER     000032  INTEGER*2 VARIABLE
FRETO   000000  REAL*4    PROCEDURE
NRECS   000034  INTEGER*2 VARIABLE
NRECSZ  000036  INTEGER*2 VARIABLE
LUN     000040  INTEGER*2 VARIABLE
J       000042  INTEGER*2 VARIABLE
FPTR    000000  REAL*4    PROCEDURE
LPTR    000044  INTEGER*2 VARIABLE
IEOF    000046  INTEGER*2 VARIABLE
IUGM    000000  INTEGER*2 PROCEDURE
FCLOSE  000000  REAL*4    PROCEDURE

COMMON BLOCK /FRECOR/   LENGTH 020000

FREECR  000000  INTEGER*2 ARRAY (4096)

COMMON BLOCK /COMTAL/   LENGTH 001000

BUF     000000  INTEGER*2 ARRAY (256)

COMMON BLOCK /NAMTAB/   LENGTH 000040

NAME    000000  LOGICAL*1 ARRAY (32)

COMMON BLOCK /INNAM/    LENGTH 000002

INAM    000000  INTEGER*2 VARIABLE

COMMON BLOCK /IGRAGU/   LENGTH 000002

IGRAGU  000000  INTEGER*2 VARIABLE
```

```
FORTRAN IV      V01C-03         MON 29-OCT-79 14:10:17          PAGE 001
CORE=08K, UIC=[1,123]                                  ,TT0:=DK3:CPIPSB

      C
0001        SUBROUTINE IRGMF
      C
      C     ,   FUNCTIONAL DESCRIPTION:
      C
      C     ,   THIS ROUTINE WILL READ THE COMTAL GRAPGICS MEMORY
      C     ,   AND WRITE IT TO A RIPS GRAPHICS (.GRA) FILE.
      C
      C
      C     ,   1) QUERY USER FOR GRAPHICS NUMBER
      C     ,   2) OPEN FILE FOR READ/WRITE
      C     ,   3) READ COMTAL, WRITE FILE
      C     ,   5) , CLOSE RIPS FILE
      C
      C
0002        INTEGER BUF(256),FREECR(4096),NAFTYP(2)
0003        LOGICAL*1 NAME(32)
0004        COMMON /OUTNAM/ONAM,/FRECOR/FREECR,/NAMTAB/NAME
0005        COMMON      /COMTAL/BUF,/IGRAQU/IGRAQU
      C
0006        DATA   NAFTYP /'.G','RA'/
      C
      C     ,   1) QUERY USER FOR GRAGHICS NUMBER
      C
0007        ICNT=1
0008        CALL        SNGDEC(IGRAQU,ICNT,IMNO,IER)
      C     ,   2)
0009        NRECS = 256
0010        NRECSZ= 32
      C
0011        CALL FCREO(ONAM,NRECS,NRECSZ,NAME,NAFTYP,LUN,BUF,IER)
      C     ,   4) READ COMTAL, WRITE TO RIPS GRAPHICS
      C
0012        DO 100 J=1,256
      C
0013        IER = IRGM (IMNO,J,FREECR)
      C
0014        CALL FPTR(LUN,FREECR,256,J,1,1,LPTR,IEOF,IER)
0015  100   CONTINUE
      C
      C
      C     ,   5) CLOSE RIPS FILE,
      C
0016        CALL FCLOSE(LUN,IER)
      C
0017        RETURN
0018        END
```

```
FORTRAN IV      STORAGE MAP

NAME    OFFSET  ATTRIBUTES

NAFTYP  000014  INTEGER*2 ARRAY (2)
ICNT    000024  INTEGER*2 VARIABLE
SNGDEC  000000  REAL*4    PROCEDURE
IMMO    000026  INTEGER*2 VARIABLE
IER     000030  INTEGER*2 VARIABLE
NRECS   000032  INTEGER*2 VARIABLE
NRECSZ  000034  INTEGER*2 VARIABLE
FCREO   000000  REAL*4    PROCEDURE
LUN     000036  INTEGER*2 VARIABLE
J       000040  INTEGER*2 VARIABLE
IRGM    000000  INTEGER*2 PROCEDURE
FPTR    000000  REAL*4    PROCEDURE
LPTR    000042  INTEGER*2 VARIABLE
IEOF    000044  INTEGER*2 VARIABLE
FCLOSE  000000  REAL*4    PROCEDURE

COMMON BLOCK /OUTNAM/   LENGTH 000004

ONAM    000000  REAL*4    VARIABLE

COMMON BLOCK /FRECOR/   LENGTH 020000

FREECR  000000  INTEGER*2 ARRAY (4096)

COMMON BLOCK /NAMTAB/   LENGTH 000040

NAME    000000  LOGICAL*1 ARRAY (32)

COMMON BLOCK /COMTAL/   LENGTH 001000

BUF     000000  INTEGER*2 ARRAY (256)

COMMON BLOCK /IGRAQU/   LENGTH 000002

IGRAQU  000000  INTEGER*2 VARIABLE
```



```
FORTRAN IV      V01C-03         MON 29-OCT-79 14:10:38          PAGE 001
COPE-02K, UIC-[1,123]                                  ,TT0:=DK3:CPIPSB

      C
0001        SUBROUTINE IWFMF
```

```
C
C     .   FUNCTIONAL DESCRIPTION:
C
C     .   THIS ROUTINE WILL READ A RIPS FUNCTION (.TRA ) FILE
C     .   AND WRITE IT TO THE COMTAL FUNCTION  MEMORY.
C
C     .   1) QUERY USER FOR FUNCTION NUMBER
C     .   2) OPEN FILE FOR READ/WRITE
C     .   3) READ FILE, WRITE COMTAL
C     .   5) , CLOSE RIPS FILE
C
0002        LOGICAL*1 NAME(32)
0003        INTEGER BUF(256),TRA(2)
0004        COMMON /FRECOR/ FREECR,/INNAM/INAM,/COMTAL/BUF
0005        COMMON /NAMTAB/NAME,/IFUNQU/IFUNQU
0006        DATA TRA /'.T','RA'/
C
C     .   1) QUERY USER FOR FUNCTION NUMBER
C
0007        ICNT=1
0008        CALL        SNGDEC(IFUNQU,ICNT,IMNO,IER)
0009        CALL FRETO(INAM,NRECS,NRECSZ,NAME,TRA,LUN,BUF,IER)
C     .   4) READ RIPS FILE, WRITE TO COMTAL FUNCTION MEMORY
C
0010        CALL FPTR(LUN,FREECR,512,1,2,0,LPTR,IEOF,IER)
C
0011        IER = IWFM (IMNO,FREECR)
C
C     .   5) CLOSE RIPS FILE,
C
C
0012        CALL FCLOSE(LUN,IER)
0013        RETURN
0014        END
```



```
FORTRAN IV      STORAGE MAP

NAME     OFFSET  ATTRIBUTES

TRA      000014  INTEGER*2 ARRAY (2)
ICNT     000030  INTEGER*2 VARIABLE
SNGDEC   000000  REAL*4    PROCEDURE
IMNO     000032  INTEGER*2 VARIABLE
IER      000034  INTEGER*2 VARIABLE
FRETO    000000  REAL*4    PROCEDURE
NRECS    000036  INTEGER*2 VARIABLE
NRECSZ   000040  INTEGER*2 VARIABLE
LUN      000042  INTEGER*2 VARIABLE
FPTR     000000  REAL*4    PROCEDURE
LPTR     000044  INTEGER*2 VARIABLE
```

```
IERF    000046  INTEGER*2 VARIABLE
IWFH    000000  INTEGER*2 PROCEDURE
FCLOSE  000000  REAL*4    PROCEDURE

COMMON BLOCK /FRECOR/   LENGTH 000004

FREECR  000000  REAL*4    VARIABLE

COMMON BLOCK /INNAM/    LENGTH 000002

INAM    000000  INTEGER*2 VARIABLE

COMMON BLOCK /COMTAL/   LENGTH 001000

BUF     000000  INTEGER*2 ARRAY (256)

COMMON BLOCK /NAMTAB/   LENGTH 000040

NAME    000000  LOGICAL*1 ARRAY (32)

COMMON BLOCK /IFUNOU/   LENGTH 000002

IFUNOU  000000  INTEGER*2 VARIABLE
```



```
FORTRAN IV      V01C-03         MON 29-OCT-79 14:11:00          PAGE 001
CORE=08K, UIC=[1,123]                                  ,TT0:=DK3:CPIPSB

        C
0001            SUBROUTINE IRFMF
        C
        C       .   FUNCTIONAL DESCRIPTION:
        C
        C       .   THIS ROUTINE WILL READ THE COMTAL FUNCTION MEMORY
        C       .   AND WRITE IT TO A RIPS TRANSFER FUNCTION (.TRA) FILE.
        C
        C       .   1) QUERY USER FOR FUNCTION NUMBER
        C       .   2) OPEN FILE FOR READ/WRITE
        C       .   3) READ COMTAL, WRITE FILE
        C       .   4) , CLOSE RIPS FILE
        C
0002            INTEGER TRA(2),FREECR(4096),BUF(256)
0003            LOGICAL*1 NAME(32)
0004            COMMON /OUTNAM/ONAM,/COMTAL/BUF,/FRECOR/FREECR,/NAMTAB/NAME
0005            COMMON/IFUNOU/IFUNOU
0006            DATA        TRA /'.T','RA'/
        C       .   1) QUERY USER FOR FUNCTION NUMBER
0007            ICNT=1
0008            CALL        SNGDEC(IFUNOU,ICNT,IMNO,IER)
0009            NRECS = 2
```

```
0010          NRECSZ= 256
              C
0011          CALL FCREO(ONAM,NRECS,NRECSZ,NAME,TRA,LUN,BUF,IER)
              C       3) READ COMTAL, WRITE TO RIPS TRANSFER FUNCTION FILE
0012          IER = IRFM (IMNO,FREECR)
              C
              C
0013          CALL FPTR(LUN,FREECR,512,1,2,1,LPTR,IEOF,IER)
              C       5) CLOSE RIPS FILE,
              C
0014          CALL FCLOSE(LUN,IER)
0015          RETURN
0016          END
```

```
FORTRAN IV      STORAGE MAP

NAME    OFFSET  ATTRIBUTES

TRA     000014  INTEGER*2 ARRAY (2)
ICNT    000026  INTEGER*2 VARIABLE
SNGDEC  000000  REAL*4    PROCEDURE
IMNO    000030  INTEGER*2 VARIABLE
IER     000032  INTEGER*2 VARIABLE
NRECS   000034  INTEGER*2 VARIABLE
NRECSZ  000036  INTEGER*2 VARIABLE
FCREO   000000  REAL*4    PROCEDURE
LUN     000040  INTEGER*2 VARIABLE
IRFM    000000  INTEGER*2 PROCEDURE
FPTR    000000  REAL*4    PROCEDURE
LPTR    000042  INTEGER*2 VARIABLE
IEOF    000044  INTEGER*2 VARIABLE
FCLOSE  000000  REAL*4    PROCEDURE

COMMON BLOCK /OUTNAM/   LENGTH 000004

ONAM    000000  REAL*4    VARIABLE

COMMON BLOCK /COMTAL/   LENGTH 001000

BUF     000000  INTEGER*2 ARRAY (256)

COMMON BLOCK /FRECOR/   LENGTH 020000

FREECR  000000  INTEGER*2 ARRAY (4096)

COMMON BLOCK /NAMTAB/   LENGTH 000040

NAME    000000  LOGICAL*1 ARRAY (32)

COMMON BLOCK /IFUNQU/   LENGTH 000002
```

```
IFCMQU  000000  INTEGER*2 VARIABLE
```

```
FORTRAN IV      V01C-03         MON 29-OCT-79 14:11:21          PAGE 001
CORE=08K, UIC=[1,123]                                    ,TT0:=DK3:CPIPSB
      C
0001        SUBROUTINE IWCBF
      C
      CC    ,   THIS ROUTINE WILL SEND ONE LINE OF TEXT
      CC    ,   TO THE COMTAL COMMAND BUFFER
      CC
      CC    ,   1) QUERY USER FOR TEXT
      C     ,   3) SEND TO COMTAL
0002        INTEGER     ISTART(1),ITTBUF(40)
0003        EQUIVALENCE (ITTBUF(2),ISTART)
0004        COMMON      /ICOMQU/ICOMQU,/TTYBUF/MSIZE,ITTBUF
      C
      CC    ,   1) QUERY USER
0005        CALL        TTYOUT(ICOMQU)
0006        CALL        TTYIN (ITTBUF)
      C     ,   3) SEND TO COMTAL
0007        IER= IWCB(ISTART)
      C     THIS FUNCTION MUST START ONN THE SECOND WORD OF THE INPUT BUFFER
      C     FOR REASONS NOT ENTIRELY CLEAR. IT IS AN IMPLICIT FIX. THE
      C     BUFFER CONTAINS LEFT OVER INFORMATION WHICH COMES FROM THE
      C     RIPS MENU PROCESSOR. NO WAY TO DATE HAS BEEN DISCOVERED TO
      C     REMOVE THE GARBAGE, SO WE JUST START 1 WORD PAST IT.
0008        RETURN
0009        END
```

```
FORTRAN IV      STORAGE MAP

NAME    OFFSET  ATTRIBUTES

TTYOUT  000000  REAL*4    PROCEDURE
TTYIN   000000  REAL*4    PROCEDURE
IER     000014  INTEGER*2 VARIABLE
IWCB    000000  INTEGER*2 PROCEDURE

COMMON BLOCK /ICOMQU/   LENGTH 000002
```

```
ICOMOU  000000  INTEGER*2 VARIABLE

(COMMON BLOCK /TTYBUF/   LENGTH 000122

MSIZE   000000  INTEGER*2 VARIABLE
ITTBUF  000002  INTEGER*2 ARRAY (40)
ISTART  000004  INTEGER*2 ARRAY (1)

FOR>
```

```
C
C
C       FILE NAME: COML01.FOR
C
C       DATE LAST EDITED: 8-MAR-79
C
C
C       HISTORY:
C
C       DATE:           MODIFICATION
C       -----           ------------
C       28-FEB-79       INITAL WRITING
C        8-MAR-79       CHANGE FUNCTION
C                       NAMES TO START WITH I
C
C                       18-SEP-79       CONVERSION TO RIPS
C
C       COMTAL LIBRARY FUNCTIONS
C       ------ ------- ---------
C
C       THIS COMTAL LIBRARY CONSISTS OF A SERIES OF FUNCTION ROUTINES
C       THE FUNCTIONS PROVIDE BASIC BUILDING BLOCKS FOR USER PROGRAMS
C       TO USE THE COMTAL.
C
C       THE SOURCE FOR THES FUNCTIONS ARE IN 5 FILES. THEY ARE:
C
C       FILE NAME               FUNCTIONS
C       ---------               ---------
C       COML01.FOR              DATA AND BIT PATTERNS FOR COMTAL
C       COML02.FOR              TRANSFER CODE 1 TYPE
C       COML03.FOR              TRANSFER CODE 2 TYPE
C       COML04.FOR              TARGET POSITION
C       COML05.FOR              TRANSFER CODE 3 TYPE
C
C       THE FOLLOWING FUNCTIONS ARE PROVIDED:
C       ------ ---- --------
C       N/A             DATA FOR COMTAL COMUNICATION
C
C        1      IWIM    WRITE   IMAGE     MEMORY
C        2              NOT     DEFINED
C        3      IRIM    READ    IMAGE     MEMORY
C        4      IWGM    WRITE   GRAPHIC   MEMORY
C        5              NOT     DEFINED
C        6      IRIM    READ    GRAPHIC   MEMORY
C        7      IWPS    WRITE   PSEUDO    MEMORY
C        8      IRPS    READ    PSEUDO    MEMORY
C        9      IWFM    WRITE   FUNCTION  MEMORY
C       10      IRFM    READ    FUNCTION  MEMORY
C       11      IWTP    WRITE   TARGET    POSITION
```

```
C     ,  12     IRTP   READ   TARGET   POSITION
C     ,  13     IWDC   WRITE  DISPLAY  COMMAND
C     ,  14            NOT    DEFINED
C     ,  15     IWCB   WRITE  COMMAND  BLOCK
C     ,  16     IRCB   READ   COMMAND  BLOCK
C     ,  17     IWCD   WRITE  CODE     DATA
C     ,  18     IRCD   READ   CODE     DATA
```



```
C     ,  19     IWDB   WRITE  DATA     BLOCK
C     ,  20     IRDB   READ   DATA     BLOCK
C
C
C     ,  ALL OF THESE ROUTINES ARE INTEGER FUNCTIONS.
C     ,  THE VALUE OF THE FUNCTION IS DETERMINED BY THE
C     ,  STATUS OF THE ROUTINE.
C
C     ,  IN GENERAL:
C
C     ,  FUNCTION VALUE          CONDITION
C     ,  -------- -----          ---------
C
C     ,           0              SUCCESS
C     ,          -1              ERROR IN READ/WRITE OF COMTAL
C
C
C
C     ***************************************************
C     ***************************************************
C     ---------------------------------------------------
C     ***************************************************
C
C
C     ,  START OF THE LIBRARY ROUTINES
0001        INTEGER FUNCTION ICMOPE
C
C
C
C
C     DATA DEFINITION
0002        INTEGER INNUMB
0003        INTEGER TCWIM, TCRIM, TCWGM, TCRGM, TCWPS, TCRPS
0004        INTEGER TCWFM, TCRFM, TCWTP, TCRTP, TCWDC, TCRDC
0005        INTEGER TCWCB, TCRCB, TCWCD, TCRCD, TCWDB, TCRDB
C
C
```

```
      C
      C     INTEGER DDB(8,20)
      C     INTEGER TCC
      C
      C     INTEGER DB1(8),DB2(8),DB3(8),DB4(8),DB5(8)
      C     INTEGER DB6(8),DB7(8),DB8(8),DB9(8),DB10(8)
      C     INTEGER DB11(8),DB12(8),DB13(8),DB14(8),DB15(8)
      C     INTEGER DB16(8),DB17(8),DB18(8),DB19(8),DB20(8)
      C
      C     LOGICAL*1 SPACE
      C
      C
      C     EQUIVALENCE (DB1 ,DDB(1,1 ))
      C     EQUIVALENCE (DB2 ,DDB(1,2 ))
```



```
      C     EQUIVALENCE (DB3 ,DDB(1,3 ))
      C     EQUIVALENCE (DB4 ,DDB(1,4 ))
      C     EQUIVALENCE (DB5 ,DDB(1,5 ))
      C     EQUIVALENCE (DB6 ,DDB(1,6 ))
      C     EQUIVALENCE (DB7 ,DDB(1,7 ))
      C     EQUIVALENCE (DB8 ,DDB(1,8 ))
      C     EQUIVALENCE (DB9 ,DDB(1,9 ))
      C     EQUIVALENCE (DB10,DDB(1,10))
      C     EQUIVALENCE (DB11,DDB(1,11))
      C     EQUIVALENCE (DB12,DDB(1,12))
      C     EQUIVALENCE (DB13,DDB(1,13))
      C     EQUIVALENCE (DB14,DDB(1,14))
      C     EQUIVALENCE (DB15,DDB(1,15))
      C     EQUIVALENCE (DB16,DDB(1,16))
      C     EQUIVALENCE (DB17,DDB(1,17))
      C     EQUIVALENCE (DB18,DDB(1,18))
      C     EQUIVALENCE (DB19,DDB(1,19))
      C     EQUIVALENCE (DB20,DDB(1,20))
      C
      C
      C     COMMON /DDB/ DDB
      C     COMMON /TC/ TCWIM, TCRIM, TCWGM, TCRGM, TCWPS, TCRPS,
     *                 TCWFM, TCRFM, TCWTP, TCRTP, TCWDC, TCRDC,
     *               TCWCB, TCRCB, TCWCD, TCRCD, TCWDB, TCRDB
      C
      C     COMMON /COMVAR/ MAXIM,MAXLN,MAXGM
      C
      C     COMMON /DATAA/ TCC
      C
      C
      C     ,   THE FOLLOWING DEFINE THE NUMBER OF CONTAL IMAGES, NUMBER OF
      C     ,   LINES INAN IMAGE, AND THE NUMBER OF GRAGHICS,
      C     ,   NOTE: ALL NUMBERS ARE 1 LESS ACTUAL
```

```
C     ,   (0 - 2, IS 3 MEMORIES)
C
0037        DATA  MAXIM /2/, MAXLN / 255 /, MAXGM / 2 /
C
C
C
C
C     ,   THE FOLLOWING TABLE DEFINES THE DDBS FOR THE
C     ,   VARIOUS FUNCTIONS DB1 IS USED FOR FUNCTION 1
C     ,   WHICH IS WRITE IMAGE MEMORY (WIM) THIS IS
C     ,   DETERMINED BY TCWIM+1.
C
0038        DATA DB1  / '101000 ,'      0 ,'0 ,'0 , -128,' 2403,' 3,' 0 /
0039        DATA DB2  / '101000,'       0 ,'0 ,'0 , - 64,' 1403,' 3,' 0 /
0040        DATA DB3  / '  1000,'  100000 ,'0 ,'0 , -128,' 2405,' 3,' 0 /
0041        DATA DB4  / '100000,'    1000 ,'0 ,'0 , - 16,' 2403,' 3,'400 /
0042        DATA DB5  / '100000,'    1000 ,'0 ,'0 , - 8,' 2403,' 3,'400 /
0043        DATA DB6  / '     0,'  101000 ,'0 ,'0 , - 16,' 2405,' 3,'400 /
0044        DATA DB7  / '140000,'   20000 ,'0 ,'0 , -256,' 3003,'23,'400 /
0045        DATA DB8  / ' 40000,'   20000 ,'0 ,'0 , -256,' 3005,' 3,'400 /
```



```
0046        DATA DB9  / '110000,'   60000 ,'0 ,'0 , -256,' 3003,'17,' 0 /
0047        DATA DB10 / ' 10000,'  160000 ,'0 ,'0 , -256,' 3005,'17,' 0 /
0048        DATA DB11 / '     0,'       0 ,'0 ,'0 , -  1,' 1403,'777,'1 /
0049        DATA DB12 / '     0,'  100000 ,'0 ,'0 , -  2,' 3405,'777,'0 /
C
0050        DATA DB13 / '     0,'       0 ,'0 ,'0 , -  1,'    3,'777,' 1 /
C
0051        DATA DB14 / '     0,'       0 ,'0 ,'0 , -  1,'  403,' 3,'400 /
0052        DATA DB15 / '150000,'   20000 ,'0 ,'0 , -256,' 3403,' 0,'  0 /
0053        DATA DB16 / ' 50000,'  120000 ,'0 ,'0 , -256,' 3405,' 0,'  0 /
0054        DATA DB17 / '110000,'   60000 ,'0 ,'0 , -256,' 3403,' 0,'  0 /
0055        DATA DB18 / ' 10000,'  160000 ,'0 ,'0 , -256,' 3405,' 0,'  0 /
0056        DATA DB19 / '140000,'   30000 ,'0 ,'0 , - 16,' 3403,' 0,'  0 /
0057        DATA DB20 / ' 40000,'  130000 ,'0 ,'0 , - 16,' 3405,' 0,'  0 /
C
C
C
0058        DATA SPACE /' '/
C
C
C     ,   THE FOLLOWING TABLE DEFINES WHICH FUNCTIONS ARE
C     ,   FUNCTION 1,2 3,ETC.
C
C     ,   EG. FUNCTION 9 IS WFM - WRITE FUNCTION MEMORY
C
```

```
      C
0053        DATA TCUIM / 1 /,  TCRIM / 3 /
0060        DATA TCUGM / 4 /,  TCRGM / 6 /
0061        DATA TCUPS / 7 /,  TCRPS / 8 /
0062        DATA TCUFM / 9 /,  TCRFM /10 /
0063        DATA TCUTP /11 /,  TCRTP /12 /
0064        DATA TCUDC /13 /,  TCRDC /14 /
0065        DATA TCUCB /15 /,  TCRCB /16 /
0066        DATA TCUCD /17 /,  TCRCD /18 /
0067        DATA TCUDB /19 /,  TCRDB /20 /
      C
      C
      C     END OF DATA DEFINITION
      C
0068        END
```

```
FORTRAN IV      STORAGE MAP

NAME    OFFSET  ATTRIBUTES

ICMOPE  000014  INTEGER*2 VARIABLE
INNUMB  000070  INTEGER*2 VARIABLE
SPACE   000016  LOGICAL*1 VARIABLE

COMMON BLOCK /DDB/      LENGTH 000500

DDB     000000  INTEGER*2 ARRAY (8,20) VECTORED
DB1     000000  INTEGER*2 ARRAY (8)
DB2     000020  INTEGER*2 ARRAY (8)
DB3     000040  INTEGER*2 ARRAY (8)
DB4     000060  INTEGER*2 ARRAY (8)
DB5     000100  INTEGER*2 ARRAY (8)
DB6     000120  INTEGER*2 ARRAY (8)
DB7     000140  INTEGER*2 ARRAY (8)
DB8     000160  INTEGER*2 ARRAY (8)
DB9     000200  INTEGER*2 ARRAY (8)
DB10    000220  INTEGER*2 ARRAY (8)
DB11    000240  INTEGER*2 ARRAY (8)
DB12    000260  INTEGER*2 ARRAY (8)
DB13    000300  INTEGER*2 ARRAY (8)
DB14    000320  INTEGER*2 ARRAY (8)
DB15    000340  INTEGER*2 ARRAY (8)
DB16    000360  INTEGER*2 ARRAY (8)
DB17    000400  INTEGER*2 ARRAY (8)
DB18    000420  INTEGER*2 ARRAY (8)
DB19    000440  INTEGER*2 ARRAY (8)
DB20    000460  INTEGER*2 ARRAY (8)

COMMON BLOCK /TC/       LENGTH 000044
```

```
TCUIM   000000  INTEGER*2 VARIABLE
TCRIM   000002  INTEGER*2 VARIABLE
TCUGM   000004  INTEGER*2 VARIABLE
TCRGM   000006  INTEGER*2 VARIABLE
TCUPS   000010  INTEGER*2 VARIABLE
TCRPS   000012  INTEGER*2 VARIABLE
TCUFM   000014  INTEGER*2 VARIABLE
TCRFM   000016  INTEGER*2 VARIABLE
TCUTP   000020  INTEGER*2 VARIABLE
TCRTP   000022  INTEGER*2 VARIABLE
TCUDC   000024  INTEGER*2 VARIABLE
TCRDC   000026  INTEGER*2 VARIABLE
TCUCB   000030  INTEGER*2 VARIABLE
TCRCB   000032  INTEGER*2 VARIABLE
TCUCD   000034  INTEGER*2 VARIABLE
TCRCD   000036  INTEGER*2 VARIABLE
TCUDB   000040  INTEGER*2 VARIABLE
TCRDB   000042  INTEGER*2 VARIABLE

COMMON BLOCK /COMVAR/   LENGTH 000006
```



```
FORTRAN IV      STORAGE MAP

NAME    OFFSET  ATTRIBUTES

MAXIM   000000  INTEGER*2 VARIABLE
MAXLM   000002  INTEGER*2 VARIABLE
MAXGM   000004  INTEGER*2 VARIABLE

COMMON BLOCK /DATAA/    LENGTH 000002

TCC     000000  INTEGER*2 VARIABLE

FOR>
```

```
FORTRAN IV      V01C-03         MON 29-OCT-79 14:13:42          PAGE 001
CORE=0SK, UIC=[1,123]                                ,TT0:=DK3:COML02.FOR

      C     ,                   FILE NAME: COML02.FOR
      C     ,
      C     ,                   DATE LAST EDITED: 8-MAR-79
      C     ,                   HISTORY:
      C     ,
      C     ,                   DATE:           MODIFICATION
      C     ,                   -----           ------------
      C     ,                   28-FEB-79       INITAL WRITING
      C     ,                    8-MAR-79       CHANGE FUNCTION
      C     ,                                   NAMES TO START WITH I
      C
      C                                    18-SEP-78      CHANGE FOR USE UNDER RIPS
0001        INTEGER FUNCTION IWIM(IMNO,IMLN,IBUF)
      C
      C     FUNCTIONAL DESCRIPTION:
      C
      C         THIS FUNCTION IS USED TO WRITE ONE LINE OF
      C         DATA FROM USER MEMORY TO THE COMTAL IMAGE SELECTED.
      C
      C     PARAMETERS:
      C
      C         IMNO  =  THE IMAGE NUMBER (1-3)
      C         IMLN  =  THE LINE NUMBER (1-256) 1 AT TOP, 256 AT BOTTOM
      C         BUFF  =  USER BUFFER ADDRESS
      C
      C     ERRORS:
      C
      C          0  =  SUCCESS
      C         -1  =  WRITE ERROR
      C
      C
      C
      C     ***************************************************
      C     DATA DEFINITION
      C
      C
0002        INTEGER TCUIM, TCRIM, TCUGM, TCRGM, TCUPS, TCRPS
0003        INTEGER TCUFM, TCRFM, TCUTP, TCRTP, TCUDC, TCRDC
0004        INTEGER TCUCD, TCRCD, TCUCD, TCRCD, TCUDB, TCRDB
      C
```

```
0005          INTEGER DDB(8,20)
0006          INTEGER TCC
      C
      C
0007          COMMON /DDB/ DDB
      C
```



```
0008          COMMON /TC/ TCWIM, TCRIM, TCWGM, TCRGM, TCWPS, TCRPS,
             *                TCWFM, TCRFM, TCWTP, TCRTP, TCWDC, TCRDC,
             *              TCWCB, TCRCB, TCWCD, TCRCD, TCWDB, TCRDB
      C
      C
      C
0009          COMMON /DATAA/ TCC
      C
      C       END OF DATA DEFINITION
      C
      C       NOTE: ALL CONSTANTS ARE DEFINED IN FUNCTION CMOPEN
      C       ----- -------------                 ------
      C
      C       ***************************************************
      C
      C
      C       ,   SET  FUNCTION CODE
      C
      C
0010          TCC=TCWIM
      C
      C
      C       ,   CALL TRANSFER CODE 1 COMMON ROUTINE
      C
0011          IWIM=ITC1(IMNO,IMLN,IBUF)
      C
0012          RETURN
0013          END
```

```
FORTRAN IV      STORAGE MAP

NAME    OFFSET  ATTRIBUTES
```

```
IRIM    000022  INTEGER*2 VARIABLE
IMNO    000014  INTEGER*2 PARAMETER VARIABLE
IMLN    000016  INTEGER*2 PARAMETER VARIABLE
IBUF    000020  INTEGER*2 PARAMETER VARIABLE
I%C1    000000  INTEGER*2 PROCEDURE

COMMON BLOCK /DDB/      LENGTH 000500

DDB     000000  INTEGER*2 ARRAY (8,20) VECTORED

COMMON BLOCK /TC/       LENGTH 000044

TCUIM   000000  INTEGER*2 VARIABLE
TCRIM   000002  INTEGER*2 VARIABLE
TCUGM   000004  INTEGER*2 VARIABLE
TCRGM   000006  INTEGER*2 VARIABLE
TCUPS   000010  INTEGER*2 VARIABLE
TCRPS   000012  INTEGER*2 VARIABLE
TCUFM   000014  INTEGER*2 VARIABLE
TCRFM   000016  INTEGER*2 VARIABLE
TCUTP   000020  INTEGER*2 VARIABLE
TCRTP   000022  INTEGER*2 VARIABLE
TCUDC   000024  INTEGER*2 VARIABLE
TCRDC   000026  INTEGER*2 VARIABLE
TCUCB   000030  INTEGER*2 VARIABLE
TCRCB   000032  INTEGER*2 VARIABLE
TCUCD   000034  INTEGER*2 VARIABLE
TCRCD   000036  INTEGER*2 VARIABLE
TCUDB   000040  INTEGER*2 VARIABLE
TCRDB   000042  INTEGER*2 VARIABLE

COMMON BLOCK /DATAA/    LENGTH 000002

TCC     000000  INTEGER*2 VARIABLE
```



```
FORTRAN IV      V01C-03         MON 29-OCT-79 14:14:05          PAGE 001
CORE=08K, UIC=[1,123]                                ,TT0:=DK3:COML02.FOR

      C
0001        INTEGER FUNCTION IRIM(IMNO,IMLN,IBUF)
      C
      C
      C     ,   FUNCTIONAL DESCRIPTION:
      C
      C     ,           THIS FUNCTION IS USED TO READ ONE LINE OF
      C     ,           DATA FROM THE COMTAL IMAGE SELECTED TO THE
      C     ,           USER BUFFER.
      C
      C     ,   PARAMETERS:
      C
```

```
C     ,          IMNO  =  THE IMAGE NUMBER (1-3)
C     ,          IMLN  =  THE LINE NUMBER (1-256) 1 AT TOP, 256 AT BOTTOM
C     ,          BUFF  =  USER BUFFER ADDRESS
C     ,
C     ,   ERRORS:
C     ,
C     ,           0  =  SUCCESS
C     ,          -1  =  WRITE ERROR
C
C
C
C     ***********************************************
C     DATA DEFINITION
C
C
0002        INTEGER TCWIM, TCRIM, TCWGM, TCRGM, TCWPS, TCRPS
0003        INTEGER TCWFM, TCRFM, TCWTP, TCRTP, TCWDC, TCRDC
0004        INTEGER TCWCB, TCRCB, TCWCD, TCRCD, TCWDB, TCRDB
C
0005        INTEGER DDB(8,20)
0006        INTEGER TCC
C
C
0007        COMMON /DDB/ DDB
C
0008        COMMON /TC/ TCWIM, TCRIM, TCWGM, TCRGM, TCWPS, TCRPS,
           1                TCWFM, TCRFM, TCWTP, TCRTP, TCWDC, TCRDC,
           2             TCWCB, TCRCB, TCWCD, TCRCD, TCWDB, TCRDB
C
C
0009        COMMON /DATAA/ TCC
C     END OF DATA DEFINITION
C
C     NOTE: ALL CONSTANTS ARE DEFINED IN FUNCTION CMOPEN
C     ----- -------------                 ------
```



```
C
C     ***********************************************************
C
C     ,   SET  FUNCTION CODE
```

```
      C
0010  C     TCC=TCRIM
      C
      C
      C     ,   CALL TRANSFER CODE 1 COMMON ROUTINE
      C
0011  C     IRIM=ITC1(IMNO,IMLN,IBUF)
      C
0012        RETURN
0013        END
```

```
FORTRAN IV      STORAGE MAP

NAME    OFFSET  ATTRIBUTES

IRIM    000022  INTEGER*2 VARIABLE
IMNO    000014  INTEGER*2 PARAMETER VARIABLE
IMLN    000016  INTEGER*2 PARAMETER VARIABLE
IBUF    000020  INTEGER*2 PARAMETER VARIABLE
ITC1    000000  INTEGER*2 PROCEDURE

COMMON BLOCK /DDB/      LENGTH 000500

DDB     000000  INTEGER*2 ARRAY (8,20) VECTORED

COMMON BLOCK /TC/       LENGTH 000044

TCWIM   000000  INTEGER*2 VARIABLE
TCRIM   000002  INTEGER*2 VARIABLE
TCWGM   000004  INTEGER*2 VARIABLE
TCRGM   000006  INTEGER*2 VARIABLE
TCWPS   000010  INTEGER*2 VARIABLE
TCRPS   000012  INTEGER*2 VARIABLE
TCWFM   000014  INTEGER*2 VARIABLE
TCRFM   000016  INTEGER*2 VARIABLE
TCWTP   000020  INTEGER*2 VARIABLE
TCRTP   000022  INTEGER*2 VARIABLE
TCWDC   000024  INTEGER*2 VARIABLE
TCRDC   000026  INTEGER*2 VARIABLE
TCWCB   000030  INTEGER*2 VARIABLE
TCRCB   000032  INTEGER*2 VARIABLE
TCWCD   000034  INTEGER*2 VARIABLE
TCRCD   000036  INTEGER*2 VARIABLE
TCWDB   000040  INTEGER*2 VARIABLE
TCRDB   000042  INTEGER*2 VARIABLE

COMMON BLOCK /DATAA/    LENGTH 000002

TCC     000000  INTEGER*2 VARIABLE
```

```
0001        INTEGER FUNCTION IWGM(IMNO,IMLN,IBUF)
C
C
C       ,   FUNCTIONAL DESCRIPTION:
C
C       ,           THIS FUNCTION IS USED TO WRITE ONE LINE OF
C       ,           DATA FROM USER MEMORY TO THE COMTAL GRAPHIC SELECTED.
C
C       ,   PARAMETERS:
C
C       ,           IMNO  =  THE GRAPHIC NUMBER (1-3)
C       ,           IMLN  =  THE LINE NUMBER (1-256) 1 AT TOP, 256 AT BOTTOM
C       ,           BUFF  =  USER BUFFER ADDRESS
C
C       ,   ERRORS:
C
C       ,            0  =  SUCCESS
C       ,           -1  =  WRITE ERROR
C
C
C
C
C       ************************************************
C       DATA DEFINITION
C
C
C
0002        INTEGER TCWIM, TCRIM, TCWGM, TCRGM, TCWPS, TCRPS
0003        INTEGER TCWFM, TCRFM, TCWTP, TCRTP, TCWDC, TCRDC
0004        INTEGER TCWCB, TCRCB, TCWCD, TCRCD, TCWDB, TCRDB
C
C
0005        INTEGER DDB(8,20)
0006        INTEGER TCC
C
C
C
0007        COMMON /DDB/ DDB
C
0008        COMMON /TC/ TCWIM, TCRIM, TCWGM, TCRGM, TCWPS, TCRPS,
           $                TCWFM, TCRFM, TCWTP, TCRTP, TCWDC, TCRDC,
           $             TCWCB, TCRCB, TCWCD, TCRCD, TCWDB, TCRDB
C
C
```

```
C
0009      COMMON /DATAA/ TCC
C
C
C     END OF DATA DEFINITION
C
C     NOTE: ALL CONSTANTS ARE DEFINED IN FUNCTION CMOPEN
C     ----- -------------                    ------
```

```
FORTRAN IV      V01C-03         MON 29-OCT-79 14:14:29          PAGE 002
CORE=0SK, UIC=[1,123]                                  ,TT0:=DK3:COML02.FOR
C
C     ************************************************************
C
C
C     ,   SET  FUNCTION CODE
C
C
0010      TCC=TCWGM
C
C
C     ,   CALL TRANSFER CODE 1 COMMON ROUTINE
C
0011      IWGM=ITC1(IMNO,IMLN,IBUF)
C
0012      RETURN
0013      END
```

```
FORTRAN IV      STORAGE MAP

NAME    OFFSET  ATTRIBUTES

IWGM    000022  INTEGER*2 VARIABLE
IMNO    000014  INTEGER*2 PARAMETER VARIABLE
IMLN    000016  INTEGER*2 PARAMETER VARIABLE
IBUF    000020  INTEGER*2 PARAMETER VARIABLE
ITC1    000000  INTEGER*2 PROCEDURE

COMMON BLOCK /DDB/      LENGTH 000500

DDB     000000  INTEGER*2 ARRAY (8,20) VECTORED

COMMON BLOCK /TC/       LENGTH 000044
```

```
[illegible]  000000  INTEGER*2 VARIABLE
[illegible]  000002  INTEGER*2 VARIABLE
[illegible]  000004  INTEGER*2 VARIABLE
[illegible]  000006  INTEGER*2 VARIABLE
[illegible]  000010  INTEGER*2 VARIABLE
[illegible]  000012  INTEGER*2 VARIABLE
[illegible]  000014  INTEGER*2 VARIABLE
[illegible]  000016  INTEGER*2 VARIABLE
[illegible]  000020  INTEGER*2 VARIABLE
[illegible]  000022  INTEGER*2 VARIABLE
[illegible]  000024  INTEGER*2 VARIABLE
[illegible]  000026  INTEGER*2 VARIABLE
[illegible]  000030  INTEGER*2 VARIABLE
[illegible]  000032  INTEGER*2 VARIABLE
[illegible]  000034  INTEGER*2 VARIABLE
[illegible]  000036  INTEGER*2 VARIABLE
[illegible]  000040  INTEGER*2 VARIABLE
[illegible]  000042  INTEGER*2 VARIABLE

COMMON BLOCK /DATAAA/    LENGTH 000002

[illegible]  000000  INTEGER*2 VARIABLE
```

```
FORTRAN IV      V01C-03          MON 29-OCT-79 14:15:00          PAGE 00:
COPE-02#, UIC-[1,123]                                  ,**0:-DK3:COMCL02.FOR
```

```
0001:       INTEGER FUNCTION IPGM(IPNO,IMLN,IBUF)

            .   FUNCTIONAL DESCRIPTION:

            .           THIS FUNCTION IS USED TO READ ONE LINE OF
            .           DATA FROM THE COMTAL GRAPHIC SELECTED TO
            .           THE USER BUFFER.

            .   PARAMETERS:

            .           IPNO  -  THE GRAPHIC NUMBER (1-3).
            .           IMLN  -  THE LINE NUMBER (1-256.) ; AT TOP, 256 AT BOTTOM
            .           BUFF  -  USER BUFFER ADDRESS

            .   ERRORS:

            .            0  -  SUCCESS
            .           -1  -  WRITE ERROR
```

```
C     ***********************************************
C
C     DATA DEFINITION
C
C
      INTEGER TCUIM, TCRIM, TCUGM, TCRGM, TCUPS, TCRPS
      INTEGER TCUFM, TCRFM, TCUTP, TCRTP, TCUDC, TCRDC
      INTEGER TCUCB, TCRCB, TCUCD, TCRCD, TCUDB, TCRDB
C
      INTEGER DDB(8,20)
      INTEGER TCC
C
C
      COMMON /DDB/ DDB
C
      COMMON /TC/ TCUIM, TCRIM, TCUGM, TCRGM, TCUPS, TCRPS,
     *            TCUFM, TCRFM, TCUTP, TCRTP, TCUDC, TCRDC,
     *            TCUCB, TCRCB, TCUCD, TCRCD, TCUDB, TCRDB
C
C
      COMMON /DATAA/ TCC
C     END OF DATA DEFINITION
C     NOTE: ALL CONSTANTS ARE DEFINED IN FUNCTION CMOPEN
```



```
C     ***********************************************
C
C     .   SET  FUNCTION CODE
C
      TCC=TCRGM
C
C     .   CALL TRANSFER CODE & COMMON ROUTINE
C
      IRGM=ITC1(IMNO,IMLN,IBUF)
C
      RETURN
      END
```

```
FORTRAN IV      STORAGE MAP

NAME    OFFSET  ATTRIBUTES

IRGM    000022  INTEGER*2 VARIABLE
IMNO    000014  INTEGER*2 PARAMETER VARIABLE
IMLN    000016  INTEGER*2 PARAMETER VARIABLE
IBUF    000020  INTEGER*2 PARAMETER VARIABLE
ITC1    000000  INTEGER*2 PROCEDURE


COMMON BLOCK /DDB/      LENGTH 000500

DDB     000000  INTEGER*2 ARRAY (8,20) VECTORED

COMMON BLOCK /TC/       LENGTH 000044

TCWIM   000000  INTEGER*2 VARIABLE
TCRIM   000002  INTEGER*2 VARIABLE
TCWGM   000004  INTEGER*2 VARIABLE
TCRGM   000006  INTEGER*2 VARIABLE
TCWPS   000010  INTEGER*2 VARIABLE
TCRPS   000012  INTEGER*2 VARIABLE
TCWFM   000014  INTEGER*2 VARIABLE
TCRFM   000016  INTEGER*2 VARIABLE
TCWTP   000020  INTEGER*2 VARIABLE
TCRTP   000022  INTEGER*2 VARIABLE
TCWDC   000024  INTEGER*2 VARIABLE
TCRDC   000026  INTEGER*2 VARIABLE
TCWCB   000030  INTEGER*2 VARIABLE
TCRCB   000032  INTEGER*2 VARIABLE
TCWCD   000034  INTEGER*2 VARIABLE
TCRCD   000036  INTEGER*2 VARIABLE
TCWDB   000040  INTEGER*2 VARIABLE
TCRDB   000042  INTEGER*2 VARIABLE

COMMON BLOCK /DATAA/    LENGTH 000002

TCC     000000  INTEGER*2 VARIABLE
```



```
FORTRAN IV      V01C-03         MON 29-OCT-79 14:15:22          PAGE 001
COPE-08K, UIC-[1,123]                                ,TT0:=DK3:COML02.FOR
```

```
0001      INTEGER FUNCTION ITC1(IMNO,IMLN,IBUF)

C
C
C     ,   FUNCTIONAL DESCRIPTION:
C     .
C     .         THIS FUNCTION IS CALLED BY ALL TRANSFER CODE 1
C     .         TYPE COMMANDS (READ/WRITE IMAGE AND GRAPHIC
C     .         MEMORIES)
C     .
C     .         THE CALLING ROUTINE SETS THE FUNCTION NUMBER
C     .         IN TCC.  THE ROUTINE CORRECTS THE LINE NUMBER,
C     .         AND IMAGE/GRAPHIC NUMBER,  FORCES THE LINE
C     .         NUMBER < 256, IMAGE NO. <3, AND THEN CALLES
C     .         THE HANDLER.
C     .
C     ,   PARAMETERS:
C     .
C     .         IMNO  =  THE IMAGE/GRAPHIC NUMBER (1-3)
C     .         IMLN  =  THE LINE NUMBER (1-256) 1 AT TOP,
C     .                  256 AT BOTTOM
C     .         BUFF  =  USER BUFFER ADDRESS
C     .
C     ,   ERRORS:
C     .
C     .          0  =  SUCCESS
C     .         -1  =  WRITE ERROR
C     .
C     .
C     .
```

```
C
C
0009        COMMON /DATAA/ TCC
C
C           END OF DATA DEFINITION
C
C           NOTE: ALL CONSTANTS ARE DEFINED IN FUNCTION CMOPEN
C           ----- -------------                      ------
C
C           ************************************************************
C
C
C           CORRECT IMAGE NUMBER, CORRECT LINE NUMBER
C
0010        NO=IMNO-1
0011        LN=IMLN-1
C
0012        MAXLN=256
0013        MAXGM=2
0014        MAXIM=2
C
C           FORCE IMAGE/GRAPHIC NUMBER, 0 <= IMAGE/GRAPHIC NO. < 2.
C           FORCE LINE NUMBER,          0 <= LINE NO.          < 256
C
0015        IF((NO.GT.MAXIM).OR.(NO.LT.0)) NO=0
0016        IF((LN.GT.MAXLN).OR.(LN.LT.0)) LN=0
C
C           SET ITC1=SUCCESS
C
0019        ITC1=0
C
C           GET USER BUFFER ADDRESS IN DDB
C
0020        DDB(4,TCC)=IADDR(IBUF)
C
C           SET TRANSFER COMMAND: SET IMAGE NUMBER, ADD LINE NUMBER
C
0021        DDB(3,TCC)=NO*4000
0022        DDB(3,TCC)=DDB(3,TCC)+LN
C
C           CALL HANDLER
C
0023        IC=IUPITW(DDB(1,TCC))
C
C           CHECK FOR ERROR
C
0024        IF(IC.LT.0) ITC1=IC
0026        RETURN
0027        END
```

```
FORTRAN IV      STORAGE MAP

NAME    OFFSET  ATTRIBUTES

ITC1    000022  INTEGER*2 VARIABLE
IMNO    000014  INTEGER*2 PARAMETER VARIABLE
IMLN    000016  INTEGER*2 PARAMETER VARIABLE
IBUF    000020  INTEGER*2 PARAMETER VARIABLE
NO      000074  INTEGER*2 VARIABLE
LN      000076  INTEGER*2 VARIABLE
MAXLN   000100  INTEGER*2 VARIABLE
MAXGM   000102  INTEGER*2 VARIABLE
MAXIM   000104  INTEGER*2 VARIABLE
IADDR   000000  INTEGER*2 PROCEDURE
IC      000106  INTEGER*2 VARIABLE
IWRITW  000000  INTEGER*2 PROCEDURE


COMMON BLOCK /DDB/      LENGTH 000500

DDB     000000  INTEGER*2 ARRAY (8,20) VECTORED

COMMON BLOCK /TC/       LENGTH 000044

TCWIM   000000  INTEGER*2 VARIABLE
TCRIM   000002  INTEGER*2 VARIABLE
TCWGM   000004  INTEGER*2 VARIABLE
TCRGM   000006  INTEGER*2 VARIABLE
TCWPS   000010  INTEGER*2 VARIABLE
TCRPS   000012  INTEGER*2 VARIABLE
TCWFM   000014  INTEGER*2 VARIABLE
TCRFM   000016  INTEGER*2 VARIABLE
TCWTP   000020  INTEGER*2 VARIABLE
TCRTP   000022  INTEGER*2 VARIABLE
TCWDC   000024  INTEGER*2 VARIABLE
TCRDC   000026  INTEGER*2 VARIABLE
TCWCB   000030  INTEGER*2 VARIABLE
TCRCB   000032  INTEGER*2 VARIABLE
TCWCD   000034  INTEGER*2 VARIABLE
TCRCD   000036  INTEGER*2 VARIABLE
TCWDB   000040  INTEGER*2 VARIABLE
TCRDB   000042  INTEGER*2 VARIABLE

COMMON BLOCK /DATAA/    LENGTH 000002

TCC     000000  INTEGER*2 VARIABLE

FOR>
```

```
C     ,                    FILE NAME: COML03.FOR
C     ,
C     ,                    DATE LAST EDITED: 21-MAR-79
C     ,
C     ,                    HISTORY:
C     ,
C     ,                    DATE:          MODIFICATION
C     ,                    -----          ------------
C     ,                    28-FEB-79      INITAL WRITING
C     ,                    19-MAR-79      CHANGE FUNCTIONS
C     ,                                   TO START WITH I
C     ,                    21-MAR-79      CHANGE ALL 'FMNO'
C     ,                                   TO 'IFMNO'
C
0001      INTEGER FUNCTION IWFM(IFMNO,IBUF)
C
C     FUNCTIONAL DESCRIPTION:
C
C         THIS ROUTINE WRITES THE SELECTED FUNCTION
C         MEMORY FROM THE USER IBUFER.
C
C     PARAMETERS:
C
C         IFMNO  =  FUNCTION MEMORY NUMBER (1-3)
C         IBUF   =  USER IBUFER (INTEGER ARRAY 256)
C
C     ERRORS:
C
C          0 =  SUCCESS
C         -1 =  WRITE  ERROR
C
C
C     ***********************************************
C
C     DATA DEFINITION
C
C
0002      INTEGER TCWIM, TCRIM, TCWGM, TCRGM, TCWPS, TCRPS
0003      INTEGER TCWFM, TCRFM, TCWTP, TCRTP, TCWDC, TCRDC
0004      INTEGER TCWCB, TCRCB, TCWCD, TCRCD, TCWDB, TCRDB
C
C
0005      INTEGER TCC
C
```

```
0006          COMMON /TC/ TCUIM, TCRIM, TCUGM, TCRGM, TCUPS, TCRPS,
     &                    TCUFM, TCRFM, TCUTP, TCRTP, TCUDC, TCRDC,
     &                TCUCB, TCRCB, TCUCD, TCRCD, TCUDB, TCRDB
      C
0007          COMMON /DATAA/ TCC
      C
      C       END OF DATA DEFINITION
```

```
FORTRAN IV      V01C-03         MON 29-OCT-79 14:16:39          PAGE 002
CORE=08K, UIC=[1,123]                                  ,TT0:=DK3:COML03.FOR

      C
      C       NOTE: ALL CONSTANTS ARE DEFINED IN FUNCTION CMOPEN
      C       ----- -------------                 ------
      C
      C       ************************************************************
      C
      C
      C      1) SET FUNCTION NUMBER
      C      2) CALL TRANSFER CODE 2 COMMON ROUTINE
      C
0008          TCC=TCUFM
      C
0009          IUFM=ITC2(IFMNO,IBUF)
      C
0010          RETURN
0011          END
```

```
FORTRAN IV      STORAGE MAP

NAME    OFFSET  ATTRIBUTES

IUFM    000020  INTEGER*2 VARIABLE
IFMNO   000014  INTEGER*2 PARAMETER VARIABLE
IBUF    000016  INTEGER*2 PARAMETER VARIABLE
ITC2    000000  INTEGER*2 PROCEDURE

COMMON BLOCK /TC/       LENGTH 000044

TCUIM   000000  INTEGER*2 VARIABLE
TCRIM   000002  INTEGER*2 VARIABLE
TCUGM   000004  INTEGER*2 VARIABLE
TCRGM   000006  INTEGER*2 VARIABLE
```

```
TCUPS   000010  INTEGER*2 VARIABLE
TCPRS   000012  INTEGER*2 VARIABLE
TCUFM   000014  INTEGER*2 VARIABLE
TCRFM   000016  INTEGER*2 VARIABLE
TCUTP   000020  INTEGER*2 VARIABLE
TCRTP   000022  INTEGER*2 VARIABLE
TCUDC   000024  INTEGER*2 VARIABLE
TCRDC   000026  INTEGER*2 VARIABLE
TCUCB   000030  INTEGER*2 VARIABLE
TCRCB   000032  INTEGER*2 VARIABLE
TCUCD   000034  INTEGER*2 VARIABLE
TCRCD   000036  INTEGER*2 VARIABLE
TCUDB   000040  INTEGER*2 VARIABLE
TCRDB   000042  INTEGER*2 VARIABLE

COMMON BLOCK /DATAA/    LENGTH 000002

TCC     000000  INTEGER*2 VARIABLE
```

```
FORTRAN IV      V01C-03         MON 29-OCT-79 14:17:02          PAGE 001
CORE=08K, UIC=[1,123]                                  .TT0:=DK3:COML03.FOR

0001        C
            INTEGER FUNCTION IRFM(IFMNO,IBUF)
            C
            C     FUNCTIONAL DESCRIPTION:
            C
            C         THIS ROUTINE READS THE SELECTED FUNCTION
            C         MEMORY AND FILLS THE USER IBUFER.
            C
            C     PARAMETERS:
            C
            C         IFMNO  =  FUNCTION MEMORY NUMBER (1-3)
            C         IBUF   =  USER IBUFER (INTEGER ARRAY 256)
            C
            C     ERROPS:
            C
            C          0 =  SUCCESS
            C         -1 =  WRITE  ERROR
            C
            C
            C     ***********************************************
            C
            C     DATA DEFINITION
            C
            C
0002        INTEGER TCUIM, TCRIM, TCUGM, TCRGM, TCUPS, TCRPS
0003        INTEGER TCUFM, TCRFM, TCUTP, TCRTP, TCUDC, TCRDC
0004        INTEGER TCUCB, TCRCB, TCUCD, TCRCD, TCUDB, TCRDB
```

```
C
C       INTEGER DDB(8,20)
0005    INTEGER TCC
0006
C
C
C
0007    C       COMMON /DDB/ DDB
0008    C       COMMON /TC/ TCUIM, TCRIM, TCUGM, TCRGM, TCUPS, TCRPS,
                *                   TCUFM, TCRFM, TCUTP, TCRTP, TCUDC, TCRDC,
                *               TCUCB, TCRCB, TCUCD, TCRCD, TCUDB, TCRDB
C
CC
0009    C       COMMON /COMUAR/ MAXIM,MAXLN,MAXGM
0010    C       COMMON /DATAA/ TCC
C
C       END OF DATA DEFINITION
C
C       NOTE: ALL CONSTANTS ARE DEFINED IN FUNCTION CMOPEN
C       ----- -------------                       ------
C
C       ***********************************************************
C
```



```
C
C       1) SET FUNCTION NUMBER
C       2) CALL TRANSFER CODE 2 COMMON ROUTINE
C
0011    C       TCC=TCRFM
0012    C       IRFM=ITC2(IFMNO,IBUF)
0013    C       RETURN
0014            END
```

FORTRAN IV STORAGE MAP

NAME OFFSET ATTRIBUTES

```
IPFM    000020  INTEGER*2 VARIABLE
[illegible] 000014  INTEGER*2 PARAMETER VARIABLE
[illegible] 000016  INTEGER*2 PARAMETER VARIABLE
[illegible] 000000  INTEGER*2 PROCEDURE

COMMON BLOCK /DDB/      LENGTH 000500

DDB     000000  INTEGER*2 ARRAY (8,20) VECTORED

COMMON BLOCK /TC/       LENGTH 000044

TCUIM   000000  INTEGER*2 VARIABLE
TCRIM   000002  INTEGER*2 VARIABLE
TCUGM   000004  INTEGER*2 VARIABLE
TCRGM   000006  INTEGER*2 VARIABLE
TCUPS   000010  INTEGER*2 VARIABLE
TCRPS   000012  INTEGER*2 VARIABLE
TCUFM   000014  INTEGER*2 VARIABLE
TCRFM   000016  INTEGER*2 VARIABLE
TCUTP   000020  INTEGER*2 VARIABLE
TCRTP   000022  INTEGER*2 VARIABLE
TCUDC   000024  INTEGER*2 VARIABLE
TCRDC   000026  INTEGER*2 VARIABLE
TCUCB   000030  INTEGER*2 VARIABLE
TCRCB   000032  INTEGER*2 VARIABLE
TCUCD   000034  INTEGER*2 VARIABLE
TCRCD   000036  INTEGER*2 VARIABLE
TCUDB   000040  INTEGER*2 VARIABLE
TCRDB   000042  INTEGER*2 VARIABLE

COMMON BLOCK /COMVAR/   LENGTH 000006

MAXIM   000000  INTEGER*2 VARIABLE
MAXLN   000002  INTEGER*2 VARIABLE
MAXGM   000004  INTEGER*2 VARIABLE

COMMON BLOCK /DATAA/    LENGTH 000002

TCC     000000  INTEGER*2 VARIABLE
```



```
FORTRAN IV      V01C-03         MON 29-OCT-79 14:17:24          PAGE 001
CORE=08K, UIC=[1,123]                                  ,TT0:=DK3:COML03.FOR

      C
0001          INTEGER FUNCTION IWPS(IBUF)
      C
      C       FUNCTIONAL DESCRIPTION:
      C
      C           THIS ROUTINE WRITES THE PSEUDO-COLOR
      C           MEMORY FROM THE USER IBUFER.
```

```
C
C     PARAMETERS:
C
C         IBUF  -  USER IBUFER (INTEGER ARRAY 256)
C
C     ERRORS:
C
C          0 :  SUCCESS
C         -1 :  WRITE  ERROR
C
C
C     ***************************************************
C
C     DATA DEFINITION
C
C
      INTEGER TCWIM, TCRIM, TCWGM, TCRGM, TCWPS, TCRPS
      INTEGER TCWFM, TCRFM, TCWTP, TCRTP, TCWDC, TCRDC
      INTEGER TCWCB, TCRCB, TCWCD, TCRCD, TCWDB, TCRDB
C
      INTEGER DDB(8,20)
      INTEGER TCC
C
      COMMON /DDB/ DDB
      COMMON /TC/ TCWIM, TCRIM, TCWGM, TCRGM, TCWPS, TCRPS,
     *            TCWFM, TCRFM, TCWTP, TCRTP, TCWDC, TCRDC,
     *            TCWCB, TCRCB, TCWCD, TCRCD, TCWDB, TCRDB
C
      COMMON /COMVAR/ MAXIM,MAXLN,MAXGM
      COMMON /DATAA/ TCC
C     END OF DATA DEFINITION
C     NOTE: ALL CONSTANTS ARE DEFINED IN FUNCTION CMOPEN
C     ---- - -------------
C     ***************************************************
```

```
C     1) SET FUNCTION NUMBER
C     2) CALL TRANSFER CODE 2 COMMON ROUTINE
C
0011        TCC=TCUPS
C
0012        IUPS=ITC2(IFMNO,IBUF)
C
0013        RETURN
0014        END
```

```
FORTRAN IV      STORAGE MAP

NAME    OFFSET  ATTRIBUTES

IUPS    000016  INTEGER*2 VARIABLE
IBUF    000014  INTEGER*2 PARAMETER VARIABLE
ITC2    000000  INTEGER*2 PROCEDURE
IFMNO   000070  INTEGER*2 VARIABLE

COMMON BLOCK /DDB/      LENGTH 000500

DDB     000000  INTEGER*2 ARRAY (8,20) VECTORED

COMMON BLOCK /TC/       LENGTH 000044

TCWIM   000000  INTEGER*2 VARIABLE
TCPIM   000002  INTEGER*2 VARIABLE
TCWGM   000004  INTEGER*2 VARIABLE
TCPGM   000006  INTEGER*2 VARIABLE
TCWPS   000010  INTEGER*2 VARIABLE
TCPPS   000012  INTEGER*2 VARIABLE
TCWFM   000014  INTEGER*2 VARIABLE
TCPFM   000016  INTEGER*2 VARIABLE
TCWTP   000020  INTEGER*2 VARIABLE
TCPTP   000022  INTEGER*2 VARIABLE
TCWDC   000024  INTEGER*2 VARIABLE
TCPDC   000026  INTEGER*2 VARIABLE
TCWCB   000030  INTEGER*2 VARIABLE
TCPCB   000032  INTEGER*2 VARIABLE
TCWCD   000034  INTEGER*2 VARIABLE
TCPCD   000036  INTEGER*2 VARIABLE
TCWDB   000040  INTEGER*2 VARIABLE
TCPDB   000042  INTEGER*2 VARIABLE

COMMON BLOCK /COMVAR/   LENGTH 000006

MAXIM   000000  INTEGER*2 VARIABLE
MAXLN   000002  INTEGER*2 VARIABLE
MAXGM   000004  INTEGER*2 VARIABLE
```

```
COMMON BLOCK /DATAA/    LENGTH 000002

TCC     000000  INTEGER*2 VARIABLE



FORTRAN IV      V01C-03         MON 29-OCT-79 14:17:56          PAGE 001
CORE-08K, UIC-[1,123]                                  ,TT0:=DK3:COML03.FOR

      C
0001        INTEGER FUNCTION IRPS(IBUF)
      C
      C     FUNCTIONAL DESCRIPTION:
      C
      C
      C         THIS ROUTINE READS THE PSEUDO-COLOR
      C         MEMORY AND PUTS IT IN THE USER IBUFER.
      C
      C     PARAMETERS:
      C
      C         IBUF  =  USER IBUFER (INTEGER ARRAY 256)
      C
      C     ERRORS:
      C
      C          0 =  SUCCESS
      C         -1 =  WRITE  ERROR
      C
      C
      C     ***************************************************
      C
      C     DATA DEFINITION
      C
      C
      C
0002        INTEGER TCWIM, TCRIM, TCWGM, TCRGM, TCWPS, TCRPS
0003        INTEGER TCWFM, TCRFM, TCWTP, TCRTP, TCWDC, TCRDC
0004        INTEGER TCWCB, TCRCB, TCWCD, TCRCD, TCWDB, TCRDB
      C
      C
0005        INTEGER DDB(8,20)
0006        INTEGER TCC
      C
      C
      C
0007        COMMON /DDB/ DDB
      C
0008        COMMON /TC/ TCWIM, TCRIM, TCWGM, TCRGM, TCWPS, TCRPS,
     1                    TCWFM, TCRFM, TCWTP, TCRTP, TCWDC, TCRDC,
     2                TCWCB, TCRCB, TCWCD, TCRCD, TCWDB, TCRDB
      C
      C
      C
0009        COMMON /DATAA/ TCC
```

```
C
C     END OF DATA DEFINITION
C     NOTE: ALL CONSTANTS ARE DEFINED IN FUNCTION CMOPEN
C     ----- -------------                  ------
C
C     **************************************************
C
C    1) SET FUNCTION NUMBER
```

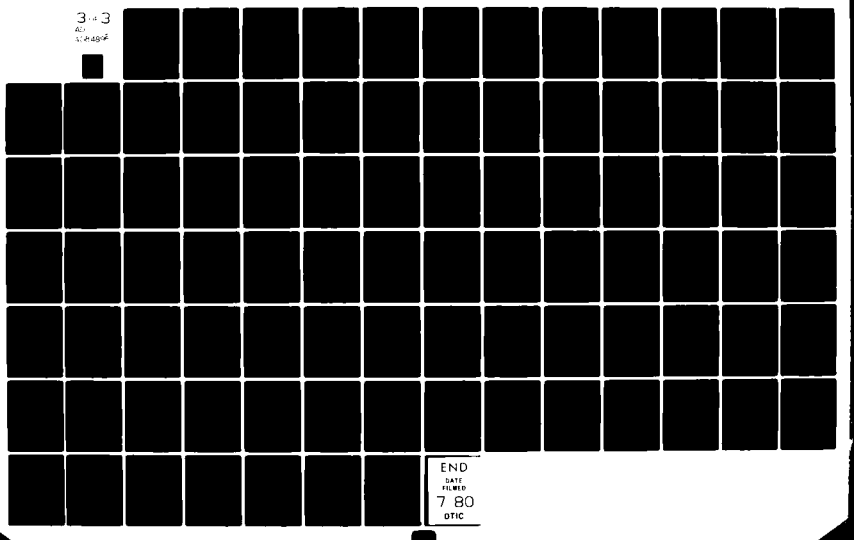
```
FORTRAN IV      V01C-03         MON 29-OCT-79 14:17:56          PAGE 002
CORE=08K, UIC=[1,123]                                ,TT0:=DK3:COML03.FOR

      C    2) CALL TRANSFER CODE 2 COMMON ROUTINE
      C
0010        TCC=TCRPS
      C
0011        IRPS=ITC2(IFMNO,IBUF)
      C
0012        RETURN
0013        END
```



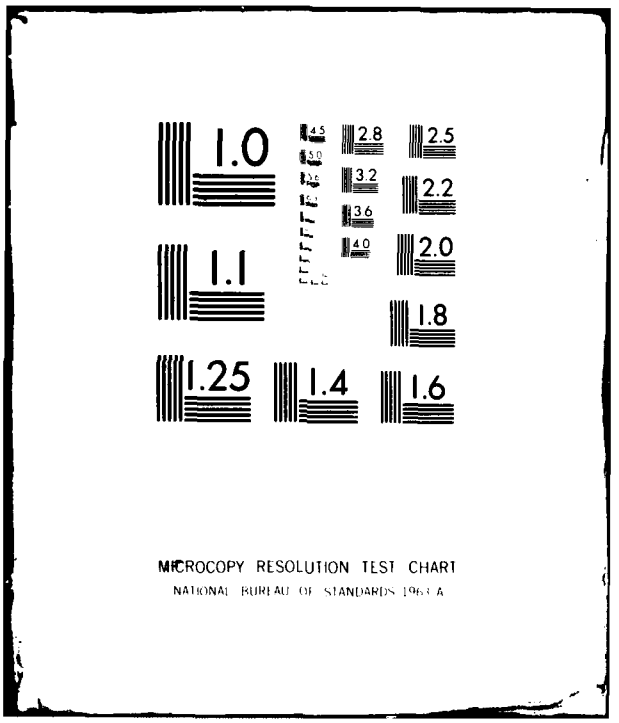
```
FORTRAN IV      STORAGE MAP

NAME    OFFSET  ATTRIBUTES

IRPS    000016  INTEGER*2 VARIABLE
IBUF    000014  INTEGER*2 PARAMETER VARIABLE
ITC2    000000  INTEGER*2 PROCEDURE
IFMNO   000070  INTEGER*2 VARIABLE

COMMON BLOCK /DDB/      LENGTH 000500

DDB     000000  INTEGER*2 ARRAY (8,20) VECTORED

COMMON BLOCK /TC/       LENGTH 000044

TCWIM   000000  INTEGER*2 VARIABLE
TCRIM   000002  INTEGER*2 VARIABLE
TCWGM   000004  INTEGER*2 VARIABLE
TCRGM   000006  INTEGER*2 VARIABLE
TCWPS   000010  INTEGER*2 VARIABLE
```

AD-A084 895 MEASUREMENT CONCEPT CORP ROME NY F/G 17/9
TACTICAL TARGET IDENTIFICATION (TTI) LABORATORY SIMULATION. VOL--ETC(U)
MAR 80 R KELLOGG, R A MCKIE, J TIMS F30602-79-C-0029
UNCLASSIFIED RADC -TR-80-81-VOL-1 NL

3 of 3
AD
A084895



END
DATE FILMED
7 80
DTIC


1.0
1.1
1.25
1.4
1.6
1.8
2.0
2.2
2.5
2.8
3.2
3.6
4.0
4.5
5.0
MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS 1963 A

```
TCPPS   000012  INTEGER*2 VARIABLE
TCSFN   000014  INTEGER*2 VARIABLE
TCRFN   000016  INTEGER*2 VARIABLE
TCWTP   000020  INTEGER*2 VARIABLE
TCRTP   000022  INTEGER*2 VARIABLE
TCWDC   000024  INTEGER*2 VARIABLE
TCRDC   000026  INTEGER*2 VARIABLE
TCWCB   000030  INTEGER*2 VARIABLE
TCRCB   000032  INTEGER*2 VARIABLE
TCWCD   000034  INTEGER*2 VARIABLE
TCRCD   000036  INTEGER*2 VARIABLE
TCWDB   000040  INTEGER*2 VARIABLE
TCRDB   000042  INTEGER*2 VARIABLE

COMMON BLOCK /DATAA/    LENGTH 000002

TCC     000000  INTEGER*2 VARIABLE
```

```
FORTRAN IV      V01C-03         MON 29-OCT-79 14:18:18          PAGE 001
CORE-08K, UIC-[1,123]                                ,TT0:-DK3:COML03.FOR

      C
0001  C     INTEGER FUNCTION ITC2(IFMNO,IBUF)
      C
      C     FUNCTIONAL DESCRIPTION:
      C
      C     THIS ROUTINE IS CALLED BY FUNCTION MEMORY , AND PSEUDO
      C     COLOR MEMORY, READ/WRITE
      C
      C
      C     CALLER SETS THE FUNCTION NUMBER
      C
      C     THE ROUTINE:
      C
      C    1) DETERMINES WHETHER PSEUDO OR FUNCTION (ONLY 1 PSEUDO MEMORY)
      C    2) FORCES FUNCTION MEMORY  < 4
      C    3) SETS COMMAND FOR TRANSFER CODE 2
      C    4) CALLS HANDLER
      C
      C     PARAMETERS:
      C
      C         IFMNO  -  FUNCTION MEMORY NUMBER (1-3)
      C         IBUF   -  USER IBUFER (INTEGER ARRAY 256)
      C
      C     ERRORS:
      C
      C          0 -  SUCCESS
      C         -1 -  WRITE  ERROR
      C
      C
      C
```

```
C     ***********************************************
C
C     DATA DEFINITION
C
C
C
0002        INTEGER TCWIM, TCRIM, TCWGM, TCRGM, TCWPS, TCRPS
0003        INTEGER TCWFM, TCRFM, TCWTP, TCRTP, TCWDC, TCRDC
0004        INTEGER TCWCB, TCRCB, TCWCD, TCRCD, TCWDB, TCRDB
C
C
0005        INTEGER DDB(8,20)
0006        INTEGER TCC
C
C
C
0007        COMMON /DDB/ DDB
C
0008        COMMON /TC/ TCWIM, TCRIM, TCWGM, TCRGM, TCWPS, TCRPS,
           $                TCWFM, TCRFM, TCWTP, TCRTP, TCWDC, TCRDC,
           $              TCWCB, TCRCB, TCWCD, TCRCD, TCWDB, TCRDB
C
C
0009        COMMON /COMVAR/ MAXIM,MAXLN,MAXGM
```



```
C
0010        COMMON /DATAA/ TCC
C
C     END OF DATA DEFINITION
C
C     NOTE: ALL CONSTANTS ARE DEFINED IN FUNCTION CMOPEN
C     ----- -------------                  ------
C     ***********************************************************
C
C
C     SET SUCCESS
C
0011        ITC2=0
C
C     CHECK FOR PSEUDO COLOR MEMORY
C
0012        IF((TCC.EQ.TCWPS).OR.(TCC.EQ.TCRPS)) GO TO 100
C
C     FUNCTION MEMORY
C
C     CORRECT FUNCTION MEMORY NUMBER
```

```
      C     LIMIT FUNCTION MEMORY NUMBER  < 4
      C
0014        NO=IFMNO-1
0015        IF((NO.GT.MAXGM).OR.(NO.LT.0)) NO=0
      C
      C     SET FUNCTION MEMORY NUMBER IN TRANSFER COMMAND
      C
0017        DDB(3,TCC)=NO*'400
      C
0018        GO TO 200
      C
      C     PSEUDO COLOR MEMORY
      C
0019  100   DDB(3,TCC)=0
      C
      C     SET IBUFER ADDRESS
      C
0020  200   DDB(4,TCC)=IADDR(IBUF)
      C
      C     CALL HANDLER
      C
0021        IC=IWRITW(DDB(1,TCC))
      C
      C     CHECK FOR ERROR
      C
0022        IF(IC.LT.0) ITC2=IC
      C
0024        RETURN
0025        END
```



```
FORTRAN IV      STORAGE MAP

NAME    OFFSET  ATTRIBUTES

ITC2    000020  INTEGER*2 VARIABLE
IFMNO   000014  INTEGER*2 PARAMETER VARIABLE
IBUF    000016  INTEGER*2 PARAMETER VARIABLE
NO      000072  INTEGER*2 VARIABLE
IADDR   000000  INTEGER*2 PROCEDURE
IC      000074  INTEGER*2 VARIABLE
IWRITW  000000  INTEGER*2 PROCEDURE

COMMON BLOCK /DDB/      LENGTH 000500

DDB     000000  INTEGER*2 ARRAY (8,20) VECTORED

COMMON BLOCK /TC/       LENGTH 000044

TCWIM   000000  INTEGER*2 VARIABLE
TCRIM   000002  INTEGER*2 VARIABLE
```

```
TCUGM    000004  INTEGER*2 VARIABLE
TCRGM    000006  INTEGER*2 VARIABLE
TCUFM    000010  INTEGER*2 VARIABLE
TCRFM    000012  INTEGER*2 VARIABLE
TCUFM    000014  INTEGER*2 VARIABLE
TCRFM    000016  INTEGER*2 VARIABLE
TCUTP    000020  INTEGER*2 VARIABLE
TCRTP    000022  INTEGER*2 VARIABLE
TCUDC    000024  INTEGER*2 VARIABLE
TCRDC    000026  INTEGER*2 VARIABLE
TCUCB    000030  INTEGER*2 VARIABLE
TCRCB    000032  INTEGER*2 VARIABLE
TCUCD    000034  INTEGER*2 VARIABLE
TCRCD    000036  INTEGER*2 VARIABLE
TCUDB    000040  INTEGER*2 VARIABLE
TCRDB    000042  INTEGER*2 VARIABLE

COMMON BLOCK /COMVAR/   LENGTH 000006

MAXIM    000000  INTEGER*2 VARIABLE
MAXLN    000002  INTEGER*2 VARIABLE
MAXGM    000004  INTEGER*2 VARIABLE

COMMON BLOCK /DATAA/    LENGTH 000002

TCC      000000  INTEGER*2 VARIABLE

FOR>
```

```
FORTRAN IV      V01C-03          MON 29-OCT-79 14:19:51          PAGE 001
CORE=08K. UIC=[1,123]                                ,TT0:=DK3:COML04.FOR

      C                        FILE NAME: COML04.FOR
      C
      C                        DATE LAST EDITED: 28-FEB-79
      C
      C
      C                        HISTORY:
      C
      C                        DATE:           MODIFICATION
      C                        -----           ------------
      C                        28-FEB-79       INITAL WRITING
      C
0001        INTEGER FUNCTION WTP(IX,IY,IXY)
      C
      C     FUNCTIONAL DESCRIPTION:
      C
      C         THIS ROUTINE WRITES X, OR Y, OR BOTH, TO THE
      C         TARGET CURSOR. (BOTH ACTUALLY REQUIRES 2 TRANSFERS)
      C
      C     PARAMETERS:
      C
      C         IX  =  X COORDINATE OF CURSOR (1-256)
      C         IY  =  Y COORDINATE OF CURSOR (1-256)
      C         IXY
      C                 =  0   TRANSFER BOTH X AND Y
      C                 =  1   TRANSFER X ONLY
      C                 =  2   TRANSFER Y ONLY
      C
      C     ERRORS:
      C
      C    0   =        SUCESS
      C   -1   =        WRITE ERROR
      C
      C
      C     ***********************************************
      C
      C     DATA DEFINITION
      C
0002        INTEGER TCWIM, TCRIM, TCWGM, TCRGM, TCWPS, TCRPS
0003        INTEGER TCWFM, TCRFM, TCWTP, TCRTP, TCWDC, TCRDC
0004        INTEGER TCWCB, TCRCB, TCWCD, TCRCD, TCWDB, TCRDB
      C
      C
0005        INTEGER DDB(8,20)
0006        INTEGER TCC
```

```
C
C
0007  C       COMMON /DDB/ DDB
C
0008  C       COMMON /TC/ TCUIM, TCRIM, TCUGM, TCRGM, TCUPS, TCRPS,
     1*                   TCUFM, TCRFM, TCUTP, TCRTP, TCUDC, TCRDC,
```



```
     1*                   TCUCB, TCRCB, TCUCD, TCRCD, TCUDB, TCRDB
C
0009  C       COMMON /DATAA/ TCC
C
C       END OF DATA DEFINITION
C
C       NOTE: ALL CONSTANTS ARE DEFINED IN FUNCTION CMOPEN
C       ----- -------------                 ------
C
C       **********************************************************
C
C
C
C       SET SUCCESS
C
0010  C       UTP=0
C
C       SET FUNCTION =URITE TARGET POSITION
C
0011  C       TCC=TCUTP
C
C       CHECK FOR X,Y,OR BOTH
C
0012  C       IF(IXY.EQ.0) GO TO 100
0014  C       IF(IXY.EQ.1) GO TO 200
C
C       JUST Y
C       SET TRANSFER COMMAND
C
C
0016  C       DDB(3,TCC)=IY
0017  C       GO TO 50
C
C
C
C       JUST X
C
0018  200   DDB(3,TCC)=IX
0019  C       GO TO 50
C
```

```
C
C
C     CALL HANDLER
0020  50    IC=IWRITW(DDB(1,TCC))
C
C     CHECK FOR ERROR
C
0021        IF(IC.LT.0) WTP=IC
0023        RETURN
      C
      C
      C
0024  100   DDB(3,TCC)=IX
```

```
FORTRAN IV      V01C-03          MON 29-OCT-79 14:19:51          PAGE 003
CORE=08K, UIC=[1,123]                                  ,TT0:=DK3:COML04.FOR

0025        IC=IWRITW(DDB(1,TCC))
0026        DDB(3,TCC)=IY
0027        ID=IWRITW(DDB(1,TCC))
      C
      C     CHECK ERROR
      C
0028        IF(IC.LT.0) WTP=IC
0030        IF(ID.LT.0) WTP=ID
      C
0032        RETURN
0033        END
```

```
FORTRAN IV      STORAGE MAP

NAME    OFFSET  ATTRIBUTES

WTP     000022  INTEGER*2 VARIABLE
IX      000014  INTEGER*2 PARAMETER VARIABLE
IY      000016  INTEGER*2 PARAMETER VARIABLE
IXY     000020  INTEGER*2 PARAMETER VARIABLE
IC      000074  INTEGER*2 VARIABLE
IWRITW  000000  INTEGER*2 PROCEDURE
ID      000076  INTEGER*2 VARIABLE

COMMON BLOCK /DDB/      LENGTH 000E00
```

```
DOF     000000  INTEGER*2 ARRAY (8,20) VECTORED

COMMON BLOCK /TC/       LENGTH 000044

TCUIM   000000  INTEGER*2 VARIABLE
TCRIM   000002  INTEGER*2 VARIABLE
TCUGM   000004  INTEGER*2 VARIABLE
TCRGM   000006  INTEGER*2 VARIABLE
TCUPS   000010  INTEGER*2 VARIABLE
TCRPS   000012  INTEGER*2 VARIABLE
TCUFM   000014  INTEGER*2 VARIABLE
TCRFM   000016  INTEGER*2 VARIABLE
TCUTP   000020  INTEGER*2 VARIABLE
TCRTP   000022  INTEGER*2 VARIABLE
TCUDC   000024  INTEGER*2 VARIABLE
TCRDC   000026  INTEGER*2 VARIABLE
TCUCB   000030  INTEGER*2 VARIABLE
TCRCB   000032  INTEGER*2 VARIABLE
TCUCD   000034  INTEGER*2 VARIABLE
TCRCD   000036  INTEGER*2 VARIABLE
TCUDB   000040  INTEGER*2 VARIABLE
TCRDB   000042  INTEGER*2 VARIABLE

COMMON BLOCK /DATAA/    LENGTH 000002

TCC     000000  INTEGER*2 VARIABLE
```



```
FORTRAN IV      V01C-03         MON 29-OCT-79 14:20:25          PAGE 001
CORE=08K, UIC=[1,123]                                  ,TT0:=DK3:COML04.FOR

      C
0001        INTEGER FUNCTION RTP(IX,IY)
      C
      C     FUNCTIONAL DESCRIPTION:
      C
      C     THIS ROUTINE READS THE X AND Y POSITION OF THE TARGET
      C     AND PUTS THE COORDINATES IN IX, AND IY .
      C
      C     PARAMETERS:
      C
      C     IX  =  X POSITION OF TARGET (1-256)
      C     IY  =  Y POSITION OF TARGET (1-256)
      C
      C     ERRORS:
      C
      C      0 =  SUCCESS
      C     -1 =  WRITE ERROR
      C
      C     ***********************************************
```

```
C
C     DATA DEFINITION
C
0002        INTEGER TAR(2)
C
C
0003        INTEGER TCWIM, TCRIM, TCWGM, TCRGM, TCWPS, TCRPS
0004        INTEGER TCWFM, TCRFM, TCWTP, TCRTP, TCWDC, TCRDC
0005        INTEGER TCWCB, TCRCB, TCWCD, TCRCD, TCWDB, TCRDB
C
C
0006        INTEGER DDB(8,20)
0007        INTEGER TCC
C
C
C
0008        COMMON /DDB/ DDB
C
0009        COMMON /TC/ TCWIM, TCRIM, TCWGM, TCRGM, TCWPS, TCRPS,
           *                TCWFM, TCRFM, TCWTP, TCRTP, TCWDC, TCRDC,
           *             TCWCB, TCRCB, TCWCD, TCRCD, TCWDB, TCRDB
C
0010        COMMON /DATAA/ TCC
C
C     END OF DATA DEFINITION
C
C     NOTE: ALL CONSTANTS ARE DEFINED IN FUNCTION CMOPEN
C     ----- -------------                  ------
C
C     **********************************************************
C
C     SET SUCCESS
C
0011        RTP=0
```



```
C
C     SET FUNCTION = READ TARGET POSITION
C
0012        TCC=TCRTP
C
C     GET ADDRESS
C
0013        DDB(4,TCC)=IADDR(TAR)
C
C     CALL HANDLER
C
0014        IC=IWRITW(DDB(1,TCC))
C
```

```
C     CHECK FOR ERRORS
C
      IF(IC.LT.0)RTP=IC
C
C     SET X, AND Y (4097 IS OFFSET FROM 1)
C
      IX=TAR(1)+4097
      IY=TAR(2)+4097
C
      RETURN
      END
```

```
FORTRAN IV      STORAGE MAP

NAME    OFFSET  ATTRIBUTES

TAR     000020  INTEGER*2 ARRAY (2)
RTP     000024  INTEGER*2 VARIABLE
IX      000014  INTEGER*2 PARAMETER VARIABLE
IY      000016  INTEGER*2 PARAMETER VARIABLE
IADDR   000000  INTEGER*2 PROCEDURE
IC      000076  INTEGER*2 VARIABLE
IURITU  000000  INTEGER*2 PROCEDURE

COMMON BLOCK /DDB/      LENGTH 000500

DDB     000000  INTEGER*2 ARRAY (8,20) VECTORED

COMMON BLOCK /TC/       LENGTH 000044

TCWIM   000000  INTEGER*2 VARIABLE
TCPIM   000002  INTEGER*2 VARIABLE
TCWGM   000004  INTEGER*2 VARIABLE
TCRGM   000006  INTEGER*2 VARIABLE
TCWPS   000010  INTEGER*2 VARIABLE
TCRPS   000012  INTEGER*2 VARIABLE
TCWFM   000014  INTEGER*2 VARIABLE
TCPFM   000016  INTEGER*2 VARIABLE
TCWTP   000020  INTEGER*2 VARIABLE
TCRTP   000022  INTEGER*2 VARIABLE
TCWDC   000024  INTEGER*2 VARIABLE
TCRDC   000026  INTEGER*2 VARIABLE
TCWCB   000030  INTEGER*2 VARIABLE
TCPCB   000032  INTEGER*2 VARIABLE
TCWCD   000034  INTEGER*2 VARIABLE
TCPCD   000036  INTEGER*2 VARIABLE
TCWDB   000040  INTEGER*2 VARIABLE
TCPDB   000042  INTEGER*2 VARIABLE

COMMON BLOCK /DATAA/    LENGTH 000002
```

T: 000000 INTEGER*2 VARIABLE
FOR

```
FORTRAN IV      V01C-03         MON 29-OCT-79 14:21:19          PAGE 001
CORE-0SK, UIC-[1,123]                                ,TT0:-DK3:COML05.FOR

      C     ,                 FILE NAME: COML05.FOR
      C     ,                 DATE LAST EDITED: 1-MAR-79
      C     ,
      C     ,                 HISTORY:
      C     ,                 DATE:           MODIFICATION
      C     ,                 -----           ------------
      C     ,                 1-MAR-79        INITAL WRITING
      C
0001        INTEGER FUNCTION IWCB(IBUF)
      C
      C     FUNCTIONAL DESCRIPTION:
      C
      C         THIS ROUTINE IS CALLED TO TRANSMIT A BLOCK OF TEXT
      C         TO THE TEXT BUFFER IN THE COMTAL.  AFTER THE TEXT
      C         IS RECEIVED, THE COMTAL WILL USE THIS TEXT AS IF IT
      C         WERE TYPED FROM THE KEYBOARD.
      C
      C     PARAMETERS:
      C
      C         BUFF  -  USER BUFFER OF TEXT
      C
      C     ERRORS:
      C
      C      0  -  SUCCESS
      C     -1  -  WRITE ERROR
      C
      C
      C     ***********************************************
      C
      C     DATA DEFINITION
      C
0002        LOGICAL*1 ITEXT(8)
      C
      C
0003        INTEGER TCWIM, TCRIM, TCWGM, TCRGM, TCUPS, TCRPS
0004        INTEGER TCWFM, TCRFM, TCWTP, TCRTP, TCWDC, TCRDC
0005        INTEGER TCWCB, TCRCB, TCWCD, TCRCD, TCWDB, TCRDB
      C
      C
0006        INTEGER DDB(8,20)
0007        INTEGER TCC
```

```
C
C
0008        COMMON /DDB/ DDB
C
0009        COMMON /TC/ TCWIM, TCRIM, TCWGM, TCRGM, TCWPS, TCRPS,
           *                TCWFM, TCRFM, TCWTP, TCRTP, TCWDC, TCRDC,
```



```
           *                TCWCB, TCRCB, TCWCD, TCRCD, TCWDB, TCRDB
C
C
C
0010        COMMON /DATAA/ TCC
C
C       END OF DATA DEFINITION
C
C       NOTE: ALL CONSTANTS ARE DEFINED IN FUNCTION CMOPEN
C       ----- -------------                         ------
C
C       **********************************************************
C
C
C       ,   1) SET COMMAND
C       ,   2) CALL COMMON CODE FOR TRANSFER CODE 3 TYPE COMMANDS
C
0011        TCC=TCWCB
C
0012        IWCB=ITC3(IBUF,IADR)
C
0013        RETURN
0014        END
```

```
FORTRAN IV      STORAGE MAP

NAME    OFFSET  ATTRIBUTES

ITE/T   000016  LOGICAL*1 ARRAY (8)
IWCB    000026  INTEGER*2 VARIABLE
IBUF    000014  INTEGER*2 PARAMETER VARIABLE
ITC3    000000  INTEGER*2 PROCEDURE
IADP    000100  INTEGER*2 VARIABLE
```

```
COMMON BLOCK /DDB/       LENGTH 000500

DDB      000000  INTEGER*2 ARRAY (8,20) VECTORED

COMMON BLOCK /TC/        LENGTH 000044

TCWIM    000000  INTEGER*2 VARIABLE
TCRIM    000002  INTEGER*2 VARIABLE
TCWGM    000004  INTEGER*2 VARIABLE
TCRGM    000006  INTEGER*2 VARIABLE
TCWPS    000010  INTEGER*2 VARIABLE
TCRPS    000012  INTEGER*2 VARIABLE
TCWFM    000014  INTEGER*2 VARIABLE
TCRFM    000016  INTEGER*2 VARIABLE
TCWTP    000020  INTEGER*2 VARIABLE
TCRTP    000022  INTEGER*2 VARIABLE
TCWDC    000024  INTEGER*2 VARIABLE
TCRDC    000026  INTEGER*2 VARIABLE
TCWCB    000030  INTEGER*2 VARIABLE
TCRCB    000032  INTEGER*2 VARIABLE
TCWCD    000034  INTEGER*2 VARIABLE
TCRCD    000036  INTEGER*2 VARIABLE
TCWDB    000040  INTEGER*2 VARIABLE
TCRDB    000042  INTEGER*2 VARIABLE

COMMON BLOCK /DATAA/     LENGTH 000002

TCC      000000  INTEGER*2 VARIABLE
```



```
FORTRAN IV      V01C-03         MON 29-OCT-79 14:21:42          PAGE 001
CORE=08K, UIC=[1,123]                                  ,TT0:=DK3:COML05.FOR

0001            C
                      INTEGER FUNCTION IRCB(IBUF)
                C
                C     FUNCTIONAL DESCRIPTION:
                C
                C         THIS ROUTINE IS CALLED TO READ THE BLOCK OF TEXT
                C         FROM THE IBUFER IN THE COMTAL.  THIS WOULD BE USED
                C         WHEN THE COMTAL INTERUPTED VIA THE 'SEND MESSAGE' REQUEST,
                C         TO READ THE TEXT TO DETERMINE WHAT THE MESSGE WAS.
                C
                C     PARAMETERS:
                C
                C         IBUF  =  USER IBUFER FOR TEXT
                C
                C     ERRORS:
                C
```

```
C       0  :  SUCCESS
C      -1  :  WRITE ERROR
C
C
C      ***************************************************
C      DATA DEFINITION
C
C
0002        INTEGER TCUIM, TCRIM, TCUGM, TCRGM, TCUPS, TCRPS
0003        INTEGER TCUFM, TCRFM, TCUTP, TCRTP, TCUDC, TCRDC
0004        INTEGER TCUCB, TCRCB, TCUCD, TCRCD, TCUDB, TCRDB
C
C
0005        INTEGER DDB(8,20)
0006        INTEGER TCC
C
C
C
0007        COMMON /DDB/ DDB
C
0008        COMMON /TC/ TCUIM, TCRIM, TCUGM, TCRGM, TCUPS, TCRPS,
           *                TCUFM, TCRFM, TCUTP, TCRTP, TCUDC, TCRDC,
           *              TCUCB, TCRCB, TCUCD, TCRCD, TCUDB, TCRDB
C
C
C
0009        COMMON /DATAA/ TCC
C
C      END OF DATA DEFINITION
C
C      NOTE: ALL CONSTANTS ARE DEFINED IN FUNCTION CMOPEN
C      ----- -------------                  ------
C
C      ***************************************************
```



```
C
C
C     1) SET COMMAND
C     2) CALL COMMON CODE FOR TRANSFER CODE 3 TYPE COMMANDS
C
0010        TCC=TCRCB
C
0011        IPCB=ITC3(IBUF,IADR)
C
0012        RETURN
```

```
0013        END


FORTRAN IV      STORAGE MAP

NAME    OFFSET  ATTRIBUTES

IRCB    000016  INTEGER*2 VARIABLE
IBUF    000014  INTEGER*2 PARAMETER VARIABLE
ITC3    000000  INTEGER*2 PROCEDURE
IADR    000070  INTEGER*2 VARIABLE


COMMON BLOCK /DDB/      LENGTH 000500

DDB     000000  INTEGER*2 ARRAY (8,20) VECTORED

COMMON BLOCK /TC/       LENGTH 000044

TCWIM   000000  INTEGER*2 VARIABLE
TCRIM   000002  INTEGER*2 VARIABLE
TCWGM   000004  INTEGER*2 VARIABLE
TCRGM   000006  INTEGER*2 VARIABLE
TCWPS   000010  INTEGER*2 VARIABLE
TCRPS   000012  INTEGER*2 VARIABLE
TCWFM   000014  INTEGER*2 VARIABLE
TCRFM   000016  INTEGER*2 VARIABLE
TCWTP   000020  INTEGER*2 VARIABLE
TCRTP   000022  INTEGER*2 VARIABLE
TCWDC   000024  INTEGER*2 VARIABLE
TCRDC   000026  INTEGER*2 VARIABLE
TCWCB   000030  INTEGER*2 VARIABLE
TCRCB   000032  INTEGER*2 VARIABLE
TCWCD   000034  INTEGER*2 VARIABLE
TCRCD   000036  INTEGER*2 VARIABLE
TCWDB   000040  INTEGER*2 VARIABLE
TCRDB   000042  INTEGER*2 VARIABLE

COMMON BLOCK /DATAA/    LENGTH 000002

TCC     000000  INTEGER*2 VARIABLE


FORTRAN IV      V01C-03         MON 29-OCT-79 14:22:04          PAGE 001
CORE=08K, UIC=[1,1233]                          ,TT0:=DK3:COML05.FOR
```

```
0001      C
          C     INTEGER FUNCTION UDB(IBUF)
          C
          C     FUNCTIONAL DESCRIPTION:
          C
          C         THIS ROUTINE IS CALLED TO TRANSMIT THE DATA BLOCK
          C         TO THE COMTAL.  THE DATA BLOCK CONTAINS THE CONTENTS
          C         OF SEVERAL DISPLAY REGISTERS THAT DETERMINE THE STATU
          C         OF THE DISPLAY.
          C
          C
          C     PARAMETERS:
          C
          C         IBUF   =  USER IBUFER OF TEXT
          C
          C     ERRORS:
          C
          C      0  =  SUCCESS
          C     -1  =  WRITE ERROR
          C
          C
          C     ***********************************************
          C
          C     DATA DEFINITION
          C
          C
          C
0002      C         INTEGER TCWIM, TCRIM, TCWGM, TCRGM, TCWPS, TCRPS
0003      C         INTEGER TCWFM, TCRFM, TCWTP, TCRTP, TCWDC, TCRDC
0004      C         INTEGER TCWCB, TCRCB, TCWCD, TCRCD, TCWDB, TCRDB
          C
          C
0005      C         INTEGER DDB(8,20)
0006      C         INTEGER TCC
          C
          C
          C
0007      C         COMMON /DDB/ DDB
          C
0008      C         COMMON /TC/ TCWIM, TCRIM, TCWGM, TCRGM, TCWPS, TCRPS,
          *                     TCWFM, TCRFM, TCWTP, TCRTP, TCWDC, TCRDC,
          *                   TCWCB, TCRCB, TCWCD, TCRCD, TCWDB, TCRDB
          C
          C
          C
0009      C         COMMON /DATAA/ TCC
          C
          C     END OF DATA DEFINITION
          C
          C
          C     NOTE: ALL CONSTANTS ARE DEFINED IN FUNCTION CMOPEN
          C     ----- -------------                 ------
          C
          C     ***********************************************************
```

```
FORTRAN IV      V01C-03         MON 29-OCT-79 14:22:04          PAGE 002
CORE-@SK, UIC-[1,123]                                  ,TT0:=DK3:COML05.FOR

      C
      C
      C     1) SET COMMAND
      C     2) CALL COMMON CODE FOR TRANSFER CODE 3 TYPE COMMANDS
      C
0010          TCC=TCUDB
      C
0011          UDB=ITC3(IBUF,IADR)
      C
0012          RETURN
0013          END



FORTRAN IV      STORAGE MAP

NAME    OFFSET  ATTRIBUTES

UDB     000016  INTEGER*2 VARIABLE
IBUF    000014  INTEGER*2 PARAMETER VARIABLE
ITC3    000000  INTEGER*2 PROCEDURE
IADR    000070  INTEGER*2 VARIABLE

COMMON BLOCK /DDB/      LENGTH 000500

DDB     000000  INTEGER*2 ARRAY (8,20) VECTORED

COMMON BLOCK /TC/       LENGTH 000044

TCUIM   000000  INTEGER*2 VARIABLE
TCPIM   000002  INTEGER*2 VARIABLE
TCUGM   000004  INTEGER*2 VARIABLE
TCRGM   000006  INTEGER*2 VARIABLE
TCUPS   000010  INTEGER*2 VARIABLE
TCPPS   000012  INTEGER*2 VARIABLE
TCUFM   000014  INTEGER*2 VARIABLE
TCRFM   000016  INTEGER*2 VARIABLE
TCUTP   000020  INTEGER*2 VARIABLE
TCRTP   000022  INTEGER*2 VARIABLE
TCUDC   000024  INTEGER*2 VARIABLE
TCPDC   000026  INTEGER*2 VARIABLE
TCUCB   000030  INTEGER*2 VARIABLE
TCPCB   000032  INTEGER*2 VARIABLE
TCUCD   000034  INTEGER*2 VARIABLE
TCRCD   000036  INTEGER*2 VARIABLE
```

```
TCWPS   000040  INTEGER*2 VARIABLE
TCRPS   000042  INTEGER*2 VARIABLE

COMMON BLOCK /DATAM.    LENGTH 000002

TCC     000000  INTEGER*2 VARIABLE



FORTRAN IV      V01C-03         MON 29-OCT-79 14:22:35          PAGE 001
CORE=08K, UIC=[1,123]                                  ,TT0:=DK3:COML05.FOR

      C
0001          INTEGER FUNCTION RDB(IBUF)
      C
      C       FUNCTIONAL DESCRIPTION:
      C
      C
      C           THIS ROUTINE IS CALLED TO READ THE COMTAL DATA BLOCK.
      C           THE DATA BLOCK CONTAINS THE TARGET POSITION, AND THE
      C           DISPLAY REGISTERS.
      C
      C
      C
      C       PARAMETERS:
      C
      C           IBUF  =  USER IBUFER OF TEXT
      C       ERRORS:
      C
      C        0  =  SUCCESS
      C       -1  =  WRITE ERROR
      C
      C
      C       ***************************************************
      C
      C       DATA DEFINITION
      C
      C
      C
0002          INTEGER TCWIM, TCRIM, TCWGM, TCRGM, TCWPS, TCRPS
0003          INTEGER TCWFM, TCRFM, TCWTP, TCRTP, TCWDC, TCRDC
0004          INTEGER TCWCB, TCRCB, TCWCD, TCRCD, TCWDB, TCRDB
      C
      C
0005          INTEGER DDB(8,20)
0006          INTEGER TCC
      C
      C
      C
0007          COMMON /DDB/ DDB
      C
0008          COMMON /TC/ TCWIM, TCRIM, TCWGM, TCRGM, TCWPS, TCRPS,
     *                    TCWFM, TCRFM, TCWTP, TCRTP, TCWDC, TCRDC,
```

```
      *              TCWCB, TCRCB, TCWCD, TCRCD, TCWDB, TCRDB
C
C
0005        COMMON /DATAA/ TCC
C
C     END OF DATA DEFINITION
C     NOTE: ALL CONSTANTS ARE DEFINED IN FUNCTION CMOPEN
C     ----- -------------                        ------
C
C     ***********************************************************
C
C    1) SET COMMAND
C    2) CALL COMMON CODE FOR TRANSFER CODE 3 TYPE COMMANDS
```

```
FORTRAN IV      V01C-03         MON 29-OCT-79 14:22:35          PAGE 002
CORE=08K, UIC=[1,123]                                  ,TT0:=DK3:COML05.FOR
C
0010        TCC=TCRDB
C
0011        RDB=ITC3(IBUF,IADR)
C
0012        RETURN
0013        END
```

```
FORTRAN IV      STORAGE MAP

NAME    OFFSET  ATTRIBUTES

RDB     000016  INTEGER*2 VARIABLE
IBUF    000014  INTEGER*2 PARAMETER VARIABLE
ITC3    000000  INTEGER*2 PROCEDURE
IADR    000070  INTEGER*2 VARIABLE

COMMON BLOCK /DDB/      LENGTH 000500

DDB     000000  INTEGER*2 ARRAY (8,20) VECTORED

COMMON BLOCK /TC/       LENGTH 000044

TCUIM   000000  INTEGER*2 VARIABLE
TCPIM   000002  INTEGER*2 VARIABLE
TCUGM   000004  INTEGER*2 VARIABLE
```

```
TCRGM   000006  INTEGER*2 VARIABLE
TCUPS   000010  INTEGER*2 VARIABLE
TCRPS   000012  INTEGER*2 VARIABLE
TCUFM   000014  INTEGER*2 VARIABLE
TCRFM   000016  INTEGER*2 VARIABLE
TCUTP   000020  INTEGER*2 VARIABLE
TCRTP   000022  INTEGER*2 VARIABLE
TCUDC   000024  INTEGER*2 VARIABLE
TCRDC   000026  INTEGER*2 VARIABLE
TCUCB   000030  INTEGER*2 VARIABLE
TCRCB   000032  INTEGER*2 VARIABLE
TCUCD   000034  INTEGER*2 VARIABLE
TCRCD   000036  INTEGER*2 VARIABLE
TCUDB   000040  INTEGER*2 VARIABLE
TCRDB   000042  INTEGER*2 VARIABLE

COMMON BLOCK /DATAA/    LENGTH 000002

TCC     000000  INTEGER*2 VARIABLE
```

```
FORTRAN IV      V01C-03         MON 29-OCT-79 14:22:57          PAGE 001
CORE=08K, UIC=[1,123]                                  ,TT0:=DK3:COML05.FOR
```

```
      C
0001        INTEGER FUNCTION IUCD(IBUF,IAD)
      C
      C     FUNCTIONAL DESCRIPTION:
      C
      C         THIS ROUTINE IS USED TO WRITE TO ANY BLOCK OF MEMORY
      C         IN THE COMTAL.  ITS PRIMARY USE IS IN DOWN LOADING CODE
      C         TO THE COMTAL, HOWEVER ANY OF THE COMTAL MEMORY LOC.
      C         CAN BE READ, AND ANY OF THE RAM AREAS WRITTEN.
      C         THE ADDRESS MUST BE ON A 256 WORD BOUNDARY.
      C
      C
      C     PARAMETERS:
      C
      C         IBUF   =  USER IBUFER OF TEXT
      C         IAD    =  ADDRESS IN THE COMTAL  FOR THE TRANSFER
      C                   (MUST BE A 512 BYTE BOUNDARY)
      C
      C     ERRORS:
      C
      C      0  =  SUCCESS
      C     -1  =  WRITE ERROR
      C
      C
      C     ********************************************
      C
      C     DATA DEFINITION
```

```
C
C
0002          INTEGER TCWIM, TCRIM, TCWGM, TCRGM, TCWPS, TCRPS
0003          INTEGER TCWFM, TCRFM, TCWTP, TCRTP, TCWDC, TCRDC
0004          INTEGER TCWCB, TCRCB, TCWCD, TCRCD, TCWDB, TCRDB
C
C
0005          INTEGER DDB(8,20)
0006          INTEGER TCC
C
C
C
0007          COMMON /DDB/ DDB
C
0008          COMMON /TC/ TCWIM, TCRIM, TCWGM, TCRGM, TCWPS, TCRPS,
             *                TCWFM, TCRFM, TCWTP, TCRTP, TCWDC, TCRDC,
             *             TCWCB, TCRCB, TCWCD, TCRCD, TCWDB, TCRDB
C
C
0009          COMMON /DATAA/ TCC
C
C     END OF DATA DEFINITION
C
C
C     NOTE: ALL CONSTANTS ARE DEFINED IN FUNCTION CMOPEN
C     ----- -------------                 ------
```



```
C
C     **********************************************************
C
C     1) SET COMMAND
C     2) CALL COMMON CODE FOR TRANSFER CODE 3 TYPE COMMANDS
C
C
0010          TCC=TCWCD
C
0011          IWCD=ITC3(IBUF,IADR)
C
0012          RETURN
0013          END
```

```
FORTRAN IV      STORAGE MAP

NAME    OFFSET  ATTRIBUTES

IUCD    000020  INTEGER*2 VARIABLE
IBUF    000014  INTEGER*2 PARAMETER VARIABLE
IAD     000016  INTEGER*2 PARAMETER VARIABLE
ITC3    000000  INTEGER*2 PROCEDURE
IADR    000072  INTEGER*2 VARIABLE


COMMON BLOCK /DDB/      LENGTH 000500

DDB     000000  INTEGER*2 ARRAY (8,20) VECTORED

COMMON BLOCK /TC/       LENGTH 000044

TCUIM   000000  INTEGER*2 VARIABLE
TCRIM   000002  INTEGER*2 VARIABLE
TCUGM   000004  INTEGER*2 VARIABLE
TCRGM   000006  INTEGER*2 VARIABLE
TCUPS   000010  INTEGER*2 VARIABLE
TCRPS   000012  INTEGER*2 VARIABLE
TCUFM   000014  INTEGER*2 VARIABLE
TCRFM   000016  INTEGER*2 VARIABLE
TCUTP   000020  INTEGER*2 VARIABLE
TCRTP   000022  INTEGER*2 VARIABLE
TCUDC   000024  INTEGER*2 VARIABLE
TCRDC   000026  INTEGER*2 VARIABLE
TCUCB   000030  INTEGER*2 VARIABLE
TCRCB   000032  INTEGER*2 VARIABLE
TCUCD   000034  INTEGER*2 VARIABLE
TCRCD   000036  INTEGER*2 VARIABLE
TCUDB   000040  INTEGER*2 VARIABLE
TCRDB   000042  INTEGER*2 VARIABLE

COMMON BLOCK /DATAA/    LENGTH 000002

TCC     000000  INTEGER*2 VARIABLE
```



```
FORTRAN IV      V01C-03         MON 29-OCT-79 14:23:20          PAGE 001
CORE=08K, UIC=[1,123]                                  ,TT0:=DK3:COML05.FOR

      C
0001        INTEGER FUNCTION IRCD(IBUF,IAD)
      C
      C     FUNCTIONAL DESCRIPTION:
      C
      C
      C         THIS ROUTINE IS USED TO READ FROM ANY BLOCK OF MEMORY
      C         IN THE CONTAL.   THE ADDRESS MUST BE ON A 256 WORD
      C         BOUNDARY.
```

```
C
C
C     PARAMETERS:
C
C         IBUF   •  USER IBUFER OF TEXT
C         IAD    •  ADDRESS IN THE COMTAL  FOR THE TRANSFER
C
C     ERRORS:
C
C      0  •  SUCCESS
C     -1  •  WRITE ERROR
C
C
C     ***************************************************
C     DATA DEFINITION
C
C
      INTEGER TCUIM, TCRIM, TCUGM, TCRGM, TCUPS, TCRPS
      INTEGER TCUFM, TCRFM, TCUTP, TCRTP, TCUDC, TCRDC
      INTEGER TCUCB, TCRCB, TCUCD, TCRCD, TCUDB, TCRDB
C
      INTEGER DDB(8,20)
      INTEGER TCC
C
C
      COMMON /DDB/ DDB
C
      COMMON /TC/ TCUIM, TCRIM, TCUGM, TCRGM, TCUPS, TCRPS,
     *                 TCUFM, TCRFM, TCUTP, TCRTP, TCUDC, TCRDC,
     *              TCUCB, TCRCB, TCUCD, TCRCD, TCUDB, TCRDB
C
C
      COMMON /DATAA/ TCC
C
C     END OF DATA DEFINITION
C
C     NOTE: ALL CONSTANTS ARE DEFINED IN FUNCTION CMOPEN
C     ----- -------------                 ------
C
```



```
C     ***************************************************
```

```
C
C
C     1) SET COMMAND
C     2) CALL COMMON CODE FOR TRANSFER CODE 3 TYPE COMMANDS
C
0010        TCC=TCRCD
C
0011        IRCD=ITC3(IBUF,IADR)
C
0012        RETURN
0013        END
```

```
FORTRAN IV      STORAGE MAP

NAME    OFFSET  ATTRIBUTES

IRCD    000020  INTEGER*2 VARIABLE
IBUF    000014  INTEGER*2 PARAMETER VARIABLE
IAD     000016  INTEGER*2 PARAMETER VARIABLE
ITC3    000000  INTEGER*2 PROCEDURE
IADR    000072  INTEGER*2 VARIABLE

COMMON BLOCK /DDB/      LENGTH 000500

DDB     000000  INTEGER*2 ARRAY (8,20) VECTORED

COMMON BLOCK /TC/       LENGTH 000044

TCWIM   000000  INTEGER*2 VARIABLE
TCRIM   000002  INTEGER*2 VARIABLE
TCWGM   000004  INTEGER*2 VARIABLE
TCRGM   000006  INTEGER*2 VARIABLE
TCWPS   000010  INTEGER*2 VARIABLE
TCRPS   000012  INTEGER*2 VARIABLE
TCWFM   000014  INTEGER*2 VARIABLE
TCRFM   000016  INTEGER*2 VARIABLE
TCWTP   000020  INTEGER*2 VARIABLE
TCRTP   000022  INTEGER*2 VARIABLE
TCWDC   000024  INTEGER*2 VARIABLE
TCRDC   000026  INTEGER*2 VARIABLE
TCWCB   000030  INTEGER*2 VARIABLE
TCRCB   000032  INTEGER*2 VARIABLE
TCWCD   000034  INTEGER*2 VARIABLE
TCRCD   000036  INTEGER*2 VARIABLE
TCWDB   000040  INTEGER*2 VARIABLE
TCRDB   000042  INTEGER*2 VARIABLE

COMMON BLOCK /DATAA/    LENGTH 000002

TCC     000000  INTEGER*2 VARIABLE
```

```
      C
      C
0001  C     INTEGER FUNCTION ITC3(IBUF,IAD)
      C
      C     FUNCTIONAL DESCRIPTION:
      C
      C         THIS IS THE COMMON ROUTINE FOR TRANSFER CODE 3
      C         TYPE COMMANDS.  IT IS CALLED BY, READ/WRITE - COMMAND
      C         BLOCK, CODE DATA, AND DATA BLOCK.
      C
      C
      C     PARAMETERS:
      C
      C         IBUF   =  USER IBUFER OF TEXT
      C         IAD    =  ADDRESS IN THE COMTAL  FOR THE TRANSFER
      C
      C     ERRORS:
      C
      C      0  =  SUCCESS
      C     -1  =  WRITE ERROR
      C
      C
      C     ***********************************************
      C
      C     DATA DEFINITION
      C
      C
0002  C     INTEGER TCWIM, TCRIM, TCWGM, TCRGM, TCWPS, TCRPS
0003  C     INTEGER TCWFM, TCRFM, TCWTP, TCRTP, TCWDC, TCRDC
0004  C     INTEGER TCWCB, TCRCB, TCWCD, TCRCD, TCWDB, TCRDB
      C
0005  C     INTEGER DDB(8,20)
0006  C     INTEGER TCC
      C
      C
0007  C     COMMON /DDB/ DDB
      C
0008  C     COMMON /TC/ TCWIM, TCRIM, TCWGM, TCRGM, TCWPS, TCRPS,
     1*                  TCWFM, TCRFM, TCWTP, TCRTP, TCWDC, TCRDC,
     2*                  TCWCB, TCRCB, TCWCD, TCRCD, TCWDB, TCRDB
      C
      C
      C
0009  C     COMMON /DATAA/ TCC
      C
```

```
C
C     END OF DATA DEFINITION
C
C     NOTE: ALL CONSTANTS ARE DEFINED IN FUNCTION CMOPEN
C     ----- -------------                              ------
C
C     ***************************************************
C
```

```
FORTRAN IV      V01C-03         MON 29-OCT-79 14:23:43          PAGE 002
CORE=08K, UIC=[1,123]                                ,TT0:=DK3:COML05.FOR

      C
0010        IA=IAD
      C
      C     SET SUCCESS
      C
0011        ITC3=0
      C
      C     GET ADDRESS
      C
0012        DDB(4,TCC)=IADDR(IBUF)
      C
0013        DDB(3,TCC)=0
      C
      C     CHECK FOR READ/WRITE OF CODE
      C
0014        IF((TCC.NE.TCWCD).AND.(TCC.NE.TCRCD)) GO TO 100
      C
      C     FUNCTION = CODE DATA, SET TRANSFER COMMAND
      C     FORCE TO EVEN 256 WORD PAGE
      C
0016        DDB(3,TCC)=(IAD/256)*256
      C
      C     CALL HANDLER
      C
0017  100   IC=IWRITW(DDB(1,TCC))
0018        IF(IC.LT.0)ITC3=IC
0020        RETURN
0021        END
```

```
FORTRAN IV      STORAGE MAP

NAME    OFFSET  ATTRIBUTES
```

```
ITI3    000020  INTEGER*2 VARIABLE
I[illegible]    000014  INTEGER*2 PARAMETER VARIABLE
I[illegible]    000016  INTEGER*2 PARAMETER VARIABLE
I[illegible]    000072  INTEGER*2 VARIABLE
I[illegible]DR  000000  INTEGER*2 PROCEDURE
IC      000074  INTEGER*2 VARIABLE
IUNITU  000000  INTEGER*2 PROCEDURE

COMMON BLOCK /DDB       LENGTH 000500

DDB     000000  INTEGER*2 ARRAY (8,20) VECTORED

COMMON BLOCK /TC/       LENGTH 000044

TCUIM   000000  INTEGER*2 VARIABLE
TCRIM   000002  INTEGER*2 VARIABLE
TCUGM   000004  INTEGER*2 VARIABLE
TCRGM   000006  INTEGER*2 VARIABLE
TCUPS   000010  INTEGER*2 VARIABLE
TCRPS   000012  INTEGER*2 VARIABLE
TCUFM   000014  INTEGER*2 VARIABLE
TCRFM   000016  INTEGER*2 VARIABLE
TCUTP   000020  INTEGER*2 VARIABLE
TCRTP   000022  INTEGER*2 VARIABLE
TCUDC   000024  INTEGER*2 VARIABLE
TCRDC   000026  INTEGER*2 VARIABLE
TCUCB   000030  INTEGER*2 VARIABLE
TCRCB   000032  INTEGER*2 VARIABLE
TCUCD   000034  INTEGER*2 VARIABLE
TCRCD   000036  INTEGER*2 VARIABLE
TCUDB   000040  INTEGER*2 VARIABLE
TCRDB   000042  INTEGER*2 VARIABLE

COMMON BLOCK /DATAA/    LENGTH 000002

TCC     000000  INTEGER*2 VARIABLE

FOR.
```

TABLE OF CONTENTS

     2-   35   IADDR   - MIMICK RT-11 SYSLIB FUNCTION IADDR
     3-   67   IURITW - MIMICK RT-11 IURITW FOR COMTAL RSX I/O


```
                                ;
                                ;                       FILE NAME:      IURITW.MAC
                                ;
                                ;                       DATE LAST CHANGE:       12-APR-79
                                ;
                                ;                                       18-SEP-79   JIM TIMS
                                ;                                                   INCORPORATE IOTRM
                                ;
                                ;                                                   SHORTEN IWPITW F
.IOTRM
                                ;
;-11 MACRO
CONVERSION
                                ;
                                ;
                                ;                       HISTORY:
                                ;
                                ;                       DATE:           MODIFICATION:
                                ;
                                ;                       11-APR-79       INITIAL WRITING
                                ;                       12-APR-79       CHANGE TCD AND FCD
                                ;       FUNCTION:
                                ;
                                ;       THIS FILE CONTAINS TWO INTEGER FUNCTIONS CALLABLE
                                ;       FROM FORTRAN.  THEY ARE:
                                ;
                                ;               IURITW - MIMICKS THE IURITW SYSLIB FUNCTION IN RT-11.
                                ;                       THIS ROUTINE WILL BE CALLED BY THE
                                ;                       COMTAL FILE UTILITY (COMPIP).
                                ;
                                ;               IADDR  - THIS ROUTINE MIMICKS THE RT-11 FUNCTION IADDR
                                ;
                                ;       NOTE: BOTH OF THESE ROUTINES WILL BE
                                ;             CALLED BY RSX-11M PROGRAMS TO
                                ;             PERFORM RT-11 SYSLIB TYPE FUNCTIONS.
```

```
                                ;
                                ;
                                ;
```

```
                                        ;
                                        ;
                                        ;       INTEGER FUNCTION IWRITW(BLOCKSIZ,DDBTABLE,BLOCKNUMBER,CHANNEL)
                                        ;
                                        ;       THIS FUNCTION CAN BE USED ONLY FOR CALLS TO THE COMTAL RSX-11M
                                        ;       HANDLER.  THE FUNCTION MIMICKS THE RT-11 SYSLIB
                                        ;       FUNCTION ' IWRITW '.  IT IS BEING WRITTEN
                                        ;       SO THAT THE COMTAL LIBRARY FOR RT-11 WILL NOT
                                        ;       REQUIRE MUCH MODIFICATION.
                                        ;
                                        ;       THE COMTAL LIBRARY HAS SEVERAL CALLS TO SYSLIB FUNCTION 'IWRITW'.
                                        ;       THESE CALLS GENERATE REQUESTS TO THE RT-11 COMTAL HANDLER.
                                        ;
                                        ;       THIS ROUTINE WILL TAKE THE DATA FROM THE IWRITW CALL AND MODIFY IT
                                        ;       TO A CALL TO QIO TO THE RSX-11M COMTAL HANDLER.
                                        ;
                                        ;       NOTE: THIS IS NOT A GENERAL PURPOSE IWRITW TO QIO CONVERSION ROUTINE.
                                        ;       IT IS TAILORED TO THE PARAMETERS USED IN THE RT-11 COMTAL HANDLER.
                                        ;
                                        ;               BLOCKSIZ    - IGNORED
                                        ;
                                        ;               DDBTABLE    - TABLE OF COMTAL PARAMETERS DEFINED AS:
                                        ;
                                        ;                       OFFSET:         PARAMETER:      DEFINITION:
                                        ;
                                        ;                       0               CL(MASK)        VALUE TO MASK CLEAR TRANS
                                        ;                       2               SET(MASK)       MASK TO SET IN TRANSFER C
                                        ;
                                        ;                       4               TC              TRANSFER COMMAND
                                        ;                       6               BSA             USER BUFFER ADDRESS
                                        ;                       10              BWC             WORD COUNT
                                        ;                       12              TCD             TRANSFER CODE (BYTE)
                                        ;                       13              FCD             READ WRITE FUNCTION (BYTE
                                        ;                                                       (3 = WRITE, 5 = READ)
                                        ;
                                        ;                       NOTE: TCD AND FDC APEAR TO HAVE THEIR DEFINITIONS REVERSE
                                        ;
                                        ;                       THE DOS HANDLER FROM COMTAL DEFINES
                                        ;                       THESE VARABLES AS ABOVE. HOWEVER, THEY USE THEM (IN THE C
                                        ;
                                        ;                       HANDLER) AS THOUGH THEIR ROLES
                                        ;                       WERE REVERSED. - RK 12-APR-79
                                                .GLOBL IWRITW
                                        ;
                        000000          CL  = 0
                        000002          SET = 2
                        000004          TC  = 4
                        000006          BSA = 6
                        000010          BWC = 10
                        000012          TCD = 12
                        000013          FCD = 13
                                        ;
000000  005725          IWRITW: TST     (R5)+           ;IGNOR NUMBER OF PARMS
                                        ;
000002  012500                  MOV     (R5)+,R0        ; GET ADDRESS OF DDBTABLE
```

```
                                                ;
                                                ;


IWRITW  MACRO M1110  29-OCT-79 14:26  PAGE 3-1
IWRITW  -  MIMICK RT-11 IWRITW FOR COMTAL RSX I/O

                                                ;               R0 NOW POINTS TO DDBTABLE
                                                ;
                                                ;
                                                ;               FIX UP TRANSFER COMMAND
                                                ;
000012  010146                          MOV     R1,-(SP)
000014  010246                          MOV     R2,-(SP)
000016  010346                          MOV     R3,-(SP)
000020  016001  000004                  MOV     TC(R0),R1       ; GET TRANSFER COMMAND
000024  046001  000000                  BIC     CL(R0),R1       ; MASK OUT PROPER BITS
000030  056001  000002                  BIS     SET(R0),R1      ; MASK IN PROPER BITS
000034  010167  000400                  MOV     R1,TCMD         ; STORE AT TRANSFER COMMAND (TCMD)
                                                ;
                                                ;               GET WORD COUNT AND FIX UP
                                                ;
000040  016001  000010                  MOV     BWC(R0),R1      ; IS NEGATIVE, NEED POSITIVE
000044  005401                          NEG     R1              ; ALSO BYTES INSTEAD OF WORDS
000046  006301                          ASL     R1
000050  010167  000360                  MOV     R1,BTCNT        ; STORE AT BYTECOUNT (BTCNT)
                                                ;
                                                ;               GET BUFFER ADDRESS
                                                ;
000054  016067  000006  000350          MOV     BSA(R0),BUFADR  ; STORE BUFFER ADDRESS (BUFADR)
                                                ;
                                                ;
                                                ;               GET TRANSFER CODE FIX UP
                                                ;
000062  116001  000013                  MOVB    FCD(R0),R1      ; MUST SHIFT AND ADD GO BIT
000066  106301                          ASLB    R1
000070  152701  000001                  BISB    #1,R1           ; ADD GO BIT
000074  010167  000336                  MOV     R1,TCODE        ; STORE AT TRANSFER CODE (TCODE)
                                                ;
                                                ;
                                                ;               DETERMINE READ OR WRITE
                                                ;
                                                ;
000100  116001  000012                  MOVB    TCD(R0),R1
000104                                  QIOW$S  #IO.INL,#18.,#52.,,#IOSTAT,
000152  122701  000003                  CMPB    #3,R1
000156  001430                          BEQ     WRIT
                                                ;
                                                                ; MUST BE READ
                                                ;
```

```
169 000160                              READIT:
170 000160                                      QIOW$S  #IO.RPP,#18.,#50.,,#IOSTAT,,<BUFADR,BTCNT,TCODE,TCMD>
171                                     ;
172 000236  000427                              BR DONE
173                                     ;
174 000240                              WRIT:   QIOW$S  #IO.WPP,#18.,#50.,,#IOSTAT,,<BUFADR,BTCNT,TCODE,TCMD>
175                                     ;
176 000316                              DONE:
177 000316                                      QIOW$S  IO.TRM,18.,52.,,#IOSTAT,
178                                     ;
179                                     ;
180 000364  016700  000036                      MOV     IOSTAT,R0               ; SUCCESS
```



```
181 000370  012603                              MOV     (SP)+,R3
182 000372  012602                              MOV     (SP)+,R2
183 000374  012601                              MOV     (SP)+,R1
184 000376  000207                              RTS     PC              ; RETURN
185                                             .GLOBL  IBYTEX
186                                     ;IBYTEX IS A BYTE EXCHANGE (SWAP) ROUTINE WHICH IS NECESSARY
187                                     ;TO INTERFACE WITH THE RIPS TTI PROGRAMS IN ORDER TO
188                                     ;REMAIN CONSISTENT WITH THE FILE FORMAT OF PREVIOUS
189                                     ;IMAGES CREATED IN THE TTI LAB.  THE BYTE STRUCTURE OF THESE
190                                     ;FILES IS NON-STANDARD FOR DEC EQUIPMENT.
191                                     ;
192                                     ;
193                                     ;COMMENCE REVERSAL
194                                     ;THE FORM FOR THE CALL FROM FORTRAN IS
195                                     ;I=IBYTEX(BUFFER,WORDCOUNT)
196 000400  016500  000002              IBYTEX: MOV     2(R5),R0        ;SET ADDRESS TO BEGIN SWAP
197 000404  017505  000004                      MOV     @4(R5),R5       ;SET    NUMBER OF WORDS TO SWAP
198 000410  010246                              MOV     R2,-(SP)        ;SAVE R2 FOR WHATEVER
199 000412  011002                      1$:     MOV     (R0),R2         ;NEXT WORD
200 000414  000302                              SWAB    R2
201 000416  010220                              MOV     R2,(R0)+
202 000420  077504                              SOB     R5,1$
203 000422  012602                              MOV     (SP)+,R2
204 000424  000207                              RTS     PC
205                                     ;
206                                     ;
207 000426                              IOSTAT: .BLKW 2         ; STATUS BLOCK
208                                     ;
209 000432  000000                      BUFADR:         .WORD 0         ; ADDRESS OF USER BUFFER
210 000434  000000                      BTCNT:          .WORD 0         ; NUMBER OF BYTES IN TRANSFER
211 000436  000000                      TCODE:          .WORD 0         ; TRANSFER CODE AND GO BIT (DRST)
212 000440  000000                      TCMD:           .WORD 0         ; TRANSFER COMMAND
213                                     ;BEGINNING HERE ARE FOUR ASCII CHARACTER STRINGS FOR USE
214                                     ;BY FUNCTIONS WHICH QUERY THE USER FOR INFORMATION.
215                                     ;THEY ARE USED BY CPIPSB AS INTEGRATED INTO THE RIPS SYSTEM
```

```
                                                ;UNDER RSX-11M
000000                                                  .PSECT  IGRAQU,RW,D,GBL,REL,OVR
000000      111     116     120                         .ASCIZ/INPUT GRAPHICS NUMBER:/
000003      125     124     040
000006      107     122     101
000011      120     110     111
000014      103     123     040
000017      116     125     115
000022      102     105     122
000025      072     000
                                                        .EVEN
000000                                                  .PSECT  IMAQUE,RW,D,GBL,REL,OVR
000000      111     116     120                         .ASCIZ/INPUT IMAGE NUMBER:/
000003      125     124     040
000006      111     115     101
000011      107     105     040
000014      116     125     115
000017      102     105     122
000022      072     000
                                                        .EVEN
000000                                                  .PSECT  IFUNQU,RW,D,GBL,REL,OVR
000000      111     116     120                         .ASCIZ/INPUT FUNCTION NUMBER:/
```



```
000003      125     124     040
000006      106     125     116
000011      103     124     111
000014      117     116     040
000017      116     125     115
000022      102     105     122
000025      072     000
                                                        .EVEN
000000                                                  .PSECT          ICOMQU,RW,D,GBL,REL,OVR
000000      105     116     124                         .ASCIZ/ENTER ONE LINE OF TEXT, END WITH $ (CR)/
000003      105     122     040
000006      117     116     105
000011      040     114     111
000014      116     105     040
000017      117     106     040
000022      124     105     130
000025      124     054     040
000030      105     116     104
000033      040     127     111
000036      124     110     040
000041      043     040     050
000044      103     122     051
000047      000
                                                        .EVEN
                                                ;
```

```
230          000001                          .END
```

```
IURITW  MACRO M1110  29-OCT-79 14:26  PAGE 3-4
SYMBOL TABLE

ESW    = 000006          DONE    000316R          IO.INL= 002400          IURITW  000006RG          TCMD    000440P
BTCNT   000434R          FCD     = 000013         IO.RPP= 001240          READIT  000160R          TCODE   000436R
BUFADR  000432R          IADDR   000000RG         IO.TRM= 002410          SET     = 000002          UPIT    000240P
BUC    = 000010          IBYTEX  000400RG         IO.UPP= 000640          TC      = 000004          $$$APR= 000002
CL     = 000000          IOSTAT  000426R          IS.SUC= 000001          TCD     = 000012

. ABS.  000000     000
        000442     001
IGRAQU  000030     002
IMAQUE  000024     003
IFUNQU  000030     004
ICOMQU  000050     005
ERRORS DETECTED:  0

VIRTUAL MEMORY USED:  1208 WORDS  ( 5 PAGES)
DYNAMIC MEMORY:  3060 WORDS  ( 11 PAGES)
ELAPSED TIME:  00:01:34
.TT0:=DK3:IURITW.MAC
MAC>
```

3.0 COMTAL: MC^2 VERSION OF THE SRC SILHOUETTE PROGRAM

The objective of this documentation is to describe FORTRAN code written to convert the SRC "SILHOUETTE" program to produce its output on the COMTAL Vision One Image Display System. Two conversions were necessary to achieve this objective:

1. Conversion from the DEC RT-11 single user real-time system to the DEC RSX-11M multi-user real-time operating system.

2. Conversion of graphic output from the calligraphic VT-11 display device to the raster format COMTAL.

The first conversion was primarily a matter of replacing some RT-11 I/O conventions with RSX conventions. As a previous task in the contract under which this work was being performed involved integration of the COMTAL under PAR's "RIPS" system, the necessary I/O capabilities for the COMTAL under RSX were already available. Data for the SILHOUETTE program was transferred from RT-11 disks to Mc^2's RSX-11 disk by means of the DEC utility "FLX".

The aircraft files were copied directly in ASCII mode. It was discovered that the parameter and image files, when transferred in "images" mode, were not readable by the RSX version of the SILHOUETTE program. It was discovered that the RT-11 FORTRAN IV "unformatted read" statements used by SRC to create these files were incompatible with the RSX-11M FORTRAN IV "unformatted read" statements. After some experimenting, it was determined that if the "unformatted write" statements in the SRC RT-11 version were replaced with "unformatted direct access write" statements, then RSX-11M would be able to read them.

A conversion program, called BOBTST.FOR was written to translate both image and parameter files from RT-11 unformatted to RT-11 direct access (unformatted). The corresponding read statements in the Mc^2

version of "SILHOUETTE" were change to direct access reads. Then "FLX" was used (image Mode) to copy the output of "BOBTST" to the Mc^2 RSX-11M disk for use by "SILHOUETTE".

Conversion of the graphic output of the SILHOUETTE program to a form acceptable to the COMTAL was a much more difficult task. Some form of lineal to raster conversion package had to be constructed to allow display of the aircraft files. The image files were somewhat simpler, requiring only coordinate system reversal (because of the COMTAL's implicit upper-left-hand-corner "origin"), translation, and 2-axis scaling, in addition to some linear expansion of each pixel's gradation. After some analysis, it was decided that because of the abbreviated time frame of the conversion effort, that program execution efficiency would be sacrificed, and that as few changes as possible would be made to the basic algorithms contained in the SRC program. Instead, Mc^2 wrote several routines which performed, in a raster environment, what the identically-named routines contained in the DEC RT-11, VT-11 FORTRAN support library performed.

The routine names and calling sequences were kept identical. Some new subroutines were written to accomplish, in software, what the VT-11 accomplished in hardware, primarily dealing with image origin and scaling. Routines supporting communication with the COMTAL under RSX-11M were task-built into the new SILHOUETTE executable task.

3.1 Operation of the New SILHOUETTE Program

3.1.1 Preparation for Execution

3.1.1.1 Conversion of Data Files

At present, only the aircraft files LEARJT.DAT, F15A.DAT and T33A.DAT have been converted from RT-11 to RSX-11M. The following instructions

should be followed to convert other (aircraft, parameter, image) file groupings that may be of interest.

- o Parameter and Image files
  - Mount Mc^2 RT-11 system packs on RK-05 drive Ø and the RT-11 data source pack on drive 1. Boot RT-11.
  - RUN DKØ:BOBTST. Respond to prompt "PARAMETER FILE NAME" by typing the full specification of the RT-11 parameter input file: i.e., DK1:T33A1P.DAT.
  - Respond to prompt "ENTER DESTINATION FILE NAME" by typing the same file name, preceded by a <u>different</u> disk prefix, in this case: DKØ:T33A1P.DAT.
  - Respond to "IMAGE FILE NAME" by typing the full specification of the image file name, i.e.: DK1:T33A1.DAT.
  - Respond to "ENTER DESTINATION FILE NAME" by typing the same file name, preceded by a different disk prefix, i.e.: DKØ:T33A1.DAT.

As it exists, the program terminates with a "STOP" after each file pair converted, and must be re-run for every conversion desired.

- o Aircraft files were written as ASCII card images, and hence do not need the conversion provided by BOBTST.FOR.
- o Transfer of files from RT-11 to RSX-11M: The RSX utility "FLX" is used to transfer files from an RT-11 disk to an RSX-11M disk. The following instructions apply.
  - Mount Mc^2 RSX-11 disk on drive Ø; RT-11 disk containing converted files on any other drive.

- Boot RSX and: SET /UIC=[1,123]

- Invoke "FLX".

- For each file triple, copy from RT-11 disk "n" to the disk labeled MCC:RSX, mounted on drive "m" as in the following example:

  -- Copy aircraft file: DKm:=DKn:T33A.DAT/RT

  -- Copy parameter file: DKm:=DKn:T33A1P.DAT/RT/IM

  -- Copy image file: DKm:=DKn:T33A1.DAT/RT/IM

3.1.2 Preparation for Execution of "SILHOUETTE" Program

3.1.2.1 Load The COMTAL Handler

The software device handler for the COMTAL must be loaded after system bootstrap, as it is normally non-resident in the RSX nucleus. Once loaded, it will remain resident until removed, or until the system is re-booted.

```
MCR>LOA ZB:
```

3.1.2.2 Install the "SILHOUETTE" Task

MCR>INS DKm:COMTAL where m is the drive number of the Mc^2:RSX disk.

3.1.2.3 Run the Task

```
MCR>RUN COMTAL
```

3.1.3 Operator Interaction with the SILHOUETTE Program

3.1.3.1 Introduction

An attempt was made to retain as much of the SRC SILHOUETTE prompting mechanism as possible. Some areas of difference exist because of the nature of the COMTAL, and an important improvement was made to the SRC version, allowing user specification of aircraft attitude in terms of yaw, pitch and roll angles, rather than the previous slant range and cross range unit orthogonal vectors, which were non-intuitive, and impossible to calculate mentally except in very special cases.

The other changes involve user specification of the COMTAL image memory to contain the resultant radar return image, and of the COMTAL graphic memory, which will contain the aircraft silhouette. There are three image memories and four graphic memories in the current Vision One, but Mc^2 programs preclude reading from or writing to the fourth graphic memory.

Translation of the aircraft silhouette, usually as a means of having it overlay the radar return image, was accomplished dynamically on the VT-11 by means of lightpen instruction. Such capability is not possible on the COMTAL becuase of the raster graphic organization, so the tracking function is replaced with a translation function that allows the user to specify how far (in pixels) the user wishes the image moved in the X and Y screen coordinate system. This translation process may be iterated as many times as desired, allowing at least the precision of the VT-11 lightpen translation.

3.1.3.2 User Interaction

```
SET  UIC=[1,123]
$ASN =B:
$INS DK3:COMTAL
$RUN COMTAL

AIRCRAFT SILHOUETTE DISPLAY
-------------------------


ENTER OPERATIONAL MODE ;
    1 = IMAGE DRIVEN MODE
    2 = STAND-ALONE MODE
1
SILHOUETTE FILE NAME

DK3:F15A.DAT

IMAGE FILE NAME

DK3:F15A1.DAT

PARAMETER FILE NAME

DK3:F15A1P.DAT
ENTER COMTAL IMAGE MEMORY NUMBER: (1-3)
3
ENTER COMTAL GRAPHIC MEMORY NUMBER (1-3):
1
TRANSLATE GRAPHIC (YES/NO)
YES
ENTER TRANSLATIONS: X,Y (REAL,SIGNED NUMBERS)
-30.,-64.
TRANSLATE GRAPHIC (YES/NO)
NO

NEW PROJECTION (YES/NO)
NO

NEW FILE (YES/NO)
NO
PROGRAM 'SILHOUETTE' TERMINATED BY OPERATOR
COMTAL -- STOP
```

Figure 3-1 Sample Command Printout

3.1.4 Building "COMTAL", The Mc^2 Version of "SILHOUETTE"

All source code, object modules, utilities and data for the "COMTAL" task reside on the disk labeled "MCC:RSX", under UIC=[1,123].

If modification to the adopted SILHOUETTE system is necessary, the user should perform the following steps:

a. Edit and recompile the affected module(s)

b. MCR>TKB @DK3:COMTAL

This second step invokes an indirect command file, which contains all information regarding the program contents and logical unit assignments of the resultant task "COMTAL".

3.2 Modification and Additions to SRC Software

3.2.1 Modifications to SLMAIN.FTN

3.2.1.1 Common Areas

GBUF - graphic airframe buffer and current X, current Y counters.
LOWLFT- X, Y coordinates of VT-11 lower screen left, and X, Y translation quantities.
LAST - X, Y coordinates of last point written to graphic image buffer.

3.2.1.2 Storage

GBUF (8192) - 256 x 256 x 1 bit graphic pixel buffer, stored in screen rows, top to bottom.

GBUF2 (32) - 256 x 1 bit graphic line buffer, for output routines.

XSHFT, YSHFT- initially zero, these are displacements of the graphic image in COMTAL frame, created by the user's previous image translation requests. This allows translations from the last graphic image location, rather than from the initial position.

3.2.1.3 Logic Changes

To convert the SILHOUETTE Program to RSX-11M, it was necessary, as previously indicated, to translate the parameter and image files from RT-11 FORTRAN unformatted I/O to direct access unformatted I/O, which is compatible with RSX. It was also necessary to insert calls to a small subroutine, FLEASN (F I L E A S S I G N), that prompts the user to enter the names of the aircraft, parameter and image files. This function was previously performed by a special option within the RT-11 "ASSIGN" subroutine, which is not present in RSX FORTRAN. "Define File" Statements were added to supply information about the parameter and image files.

As an improvement to the original SRC method of specifying aircraft attitude in terms of unit slant range and cross range vectors, Mc^2 wrote a subroutine called "ROTATE" that accepts yaw, pitch and roll angles, and constructs the unit vectors from this information.

The VT-11 light pen tracking routine, which allowed the user to superimpose the aircraft image over the radar return image, has been replaced by a prompt requesting the user to enter, if he wishes to translate the image, the signed displacement of the aircraft image that is desired. This translation is measured in pixels and is referenced to the lower left corner of the screen. The COMTAL MACRO routine "INS" can be used to determine the translation values.

Variable SSZ, screen size in inches, was changed from 12. to 10.5, the visible dimension of the COMTAL.

3.2.1.4 Known Problems

3.2.1.4.1 Byte Exchange

The COMTAL supplied RSX-11 device handler does not swap bytes within a PDP-11 word during a transfer, as its RT-11 predecessor did. For this reason, routine IBYTEX is called to place the pixels in proper logical order before transfer.

3.2.1.4.2 Specification of a New File While in Image-Driven Mode, (Mode 1)

Due to some quirk of FORTRAN, attempts to close a parameter and image file and re-open a new set of parameter and image files results in a file name specification error; that could not be fully understood because of missing RSX-11M FORTRAN documentation. As a temporary measure, the statement at line #47 will bypass this problem by directing all requests for "new file" while in Mode 1 to a graceful exit: the new file can be used by re-running "COMTAL".

3.2.2 Modification to SLSUBS.FTN

SLSUBS is a collection of FORTRAN subroutines, written by SRC, that has been modified as little as possible for use in the COMTAL version of SILHOUETTE. One routine, FLEASN, was added to read file names for the ASSIGN calls in SLMAIN.

The following COMMON areas and variables apply to one or more members of SLSUBS. Logic changes will be discusson a routine-by-routine basis.

3.2.2.1 Common Areas

SCALE - holds scaling factor for aircraft SILHOUETTE construction.
LAST - See SLMAIN.
LOWLFT - See SLMAIN.
GBUF - See SLMAIN.

3.2.2.2 Storage

GBUS (8192) - Graphic pixel buffer.
COORD(6) - A temporary buffer for holding data values and applying the scale factor (SCL) to them.
D(256) - Image line buffer, input (real data).
ITEXT (8) - LOGICAL*1 array to store "CLEAR IMAGE BUFFER" command.
IMLINE - 256-byte image line output buffer.

3.2.2.3 Logic Changes

3.2.2.3.1 Subroutine Draw

Subroutine ADJUST, which multiplies all VT-11 coordinate positions by the scale factor SCL computed by Subroutine SCALE, appears directly before calls to routines writing points, lines and circles. This routine simulates the VT-11 hardware scaling feature.

3.2.2.3.2 Subroutine CIRCLE

No changes.

3.2.2.3.3 Subroutine CYLNDR

Other than call to ADJUST at Line #10, no changes.

3.2.2.3.4 Subroutine PASS2

@ Lines #35-40, prompt for specification of image buffer to be used.
@ Lines #41-47, clears proper COMTAL image buffer.
@ Line #66, image intensity level is extended to 8 bits from the 3 bits of the VT-11.
@ Lines #72-75, write a line of the image to the COMTAL.

3.2.2.3.5 Subroutine TVECT

No changes.

3.2.2.3.6 Subroutine PLOT

No changes.

3.2.2.3.7 Subroutine FLEASN

Allows user to enter filenames of aircraft, parameter and image files.

3.2.3 COMGRF.FTN

COMGRF is a set of FORTRAN routines that emulate on the COMTAL routines of the same name in the DEC VT-11 support library. These routines allow most of the other SILHOUETTE software to be very similar to the RT-11 version.

3.2.3.1 Subroutine SCAL (Min x, Min y, Max x, Max y)

Determines the spread between (Min x, Max x) and (Min y, Max y) and finds the larger difference, which is then divided by 1024 to get the VT-11 distance per unit increment. This is done because the COMTAL is a square display, rather than rectangular. The coordinates of the lower left corner of the display are determined by adding together 512 (to convert from the VT-11 SILHOUETTE convention of a center-screen origin) and four times the desired pixel displacement.

3.2.3.2 Subroutine APNT

Resets the current x, current y counters to the x, y specified in the argument list. Currently unused by SILHOUETTE.

3.2.3.3 Subroutine RDOT (DX, DY)

Moves the current x, y pointers to (CX+DX, CY+DY) and calls CDOT to illuminate that pixel.

3.2.3.4 Subroutine LVECT (DX, DY)

LVECT is a lineal to raster conversion routine that "draws" a straight line in the graphic buffer between (CX, CY) and (CX+DX, CY+DY). LVECT determines the slope of this line. If the absolute value of the slope is less than 1, then the line is described by a formula of the form $y = f(x)$. Conversely, if the absolute value of the slope is >1, then the line is described in the form $x=f(y)$.

After the line's equation has been calculated, discrete values of the equations dependent variable are calculated by stepping the independent variable between its input limits in unit intervals. As the dependent variable is rarely of integer value, it is rounded up or down to its nearest integral value. Subroutine CDOT is called to place each (integral

independent variable, dependent variable) pair in the graphic display buffer. When all such pairs have been sent, the current x and current y counters are updated to reflect the end of the line.

3.2.3.5 Subroutine CDOT (x, y)

Subroutine CDOT accepts pixel addresses in the VT-11 1024 x 1024 coordinates frame (Lower Left Origin), and converts these to pixels in the COMTAL 256 x 256 upper left origin frame. It then calculates the bit address of the pixel in array "GBUF" and sets this bit.

Quantities XLAST and YLAST contain the last x, y position set in the COMTAL buffer. If a request to set the same bit as set in the last call to CDOT (which often happens because of the 4:1 coordinate frame shrinkage), the request is ignored. Otherwise, XLAST, YLAST is reset to reflect the current request and the bits in GBUF are set.

3.2.3.6 Subroutine ADJUST

This routine accepts an array of real variables and a counter N, and applies the multiplicative scale factor SCL to the first N values of the array. This allows one routine to suffice for line and point requests, which have three parameters requiring scaling, and the circle requests, which have four scalable parameters.

3.2.4 Subroutine ROTATE

Subroutine ROTATE accepts three real, signed input parameters, which correspond to yaw, pitch and roll angles.

Yaw is measured clockwise if looking down onto top of aircraft. Pitch is measured positive if aircraft nose is pointed above horizontal, negative below. Roll is measured position clockwise when looking down the aircraft fuselage.

The input values, in signed degrees, are converted to radians, and unit vectors RS and RC are calculated from these values.

3.2.5 VECTOR.FTN

The three vector algebra routines UNITV, DOTP and CROSSP are used without change from their RT-11 versions.

3.2.6 Other Programs

3.2.6.1 BOBTST.FTN

This stand-alone RT-11 program assists the user in copying RT-11 FORTRAN unformatted files to RT-11 direct-access format. The direct access format is compatible with RSX-11M FORTRAN, and can be "FLX'ed" from an RT-11 pack to an RSX-11M pack for use with the SILHOUETTE Program.

It is necessary to specify different disk packs for source and destination files when running this program so that the original file name can be retained. This is important becuase FLX will not support the renaming of files during transfer.

3.2.6.2 COMTAL.CMD

This file is the task builder indirect command file that builds the task COMTAL, which is the Mc^2 version of the SRC task SILUET.SAV.

```
FORTRAN IV      V01C-03         FRI 28-SEP-79 09:28:22          PAGE 001
CORE=08K, UIC=[1,123]                                ,TT0:=DK3:SLMAIN.FTN

C           AIRCRAFT SILHOUETTE DISPLAY PROGRAM
C                 P.A.D.C.   -   PDP 11-40
C
C
C          MODIFIED BY ROBERT A. MCKIE, MEASUREMENT CONCEPT CORP.,
C          TO SUPPORT COMTAL VISION ONE
C          SEPTEMBER, 1979
C
C
C
C
C           PROGRAM FILES
C           -------------
C
C               LOAD MODULE NAME            = COMTAL.SAV
C
C               SOURCE FILES
C
C                   MAIN PROGRAM            = SLMAIN.FTN
C                   DRAW,CIRCLE,CYLNDR,
C                   PASS2,TVECT,TRACK,PLOT  = SLSUBS.FTN
C                   UNITV,DOTP,CROSSP       = VECTOR.FTN
C                   SCAL,APNT,RDOT,LVECT,
C                   CDOT,ADJUST             = COMGRF.FTN
C                   ROTATE                  = ROTATE.FTN
C
C               OBJECT FILES
C
C                   MAIN PROGRAM            = SLMAIN.OBJ
C                   DRAW,CIRCLE,CYLNDR,
C                   PASS2,TVECT,TRACK,PLOT  = SLSUBS.OBJ
C                   UNITV,DOTP,CROSSP       = VECTOR.OBJ
C                   GRAPHIC ROUTINES        = COMGRF.OBJ
C                   YAW,PITCH,ROLL ENTRY    = ROTATE.OBJ
C
C
0001          COMMON /GBUF/GBUF,CX,CY
0002          COMMON /LOWLFT/ XLL,YLL,XSHFT,YSHFT
0003          COMMON /LAST/ XLAST,YLAST
0004          LOGICAL*1 GBUF(8192)
0005          LOGICAL*1 GBUF2(32)
0006          INTEGER AFNAME(7),IMFILE(7),PARAFL(7)
```

```
0004          INTEGER AIRCFL,TTYI,TTYO
0005          REAL RS(3),RC(3),RZ(3),FINVIS(2)
0006          INTEGER*2 XLAST,YLAST
0010          DATA AIRCFL/9/,NIMG/10/,NPRM/11/
0011          DATA TTYI/5/,TTYO/7/,NO/'NO '/
0012          DATA DTR/.01745329/,SSZ/10.5/,SDT/1024./,EPS/.05/
0013          DATA NCRI/10/
```

```
FORTRAN IV      V01C-03         FRI 28-SEP-79 09:28:22          PAGE 002
CORE=08K, UIC=[1,123]                                  ,TT0:=DK3:SLMAIN.FTN

0014          DATA DMIN/-100./,DBTHR/-20./
C
C
C
C             DEFINE FILES FOR PARAMETERS, IMAGE
C
C
C
C              INITIAL INPUT PARAMETERS
C              ------------------------
C
C                  MODE  = OPERATIONAL MODE (1 OR 2)
C                  NAME  = DATA FILE NAME (10 CHARACTERS)
C                  SCALE = SCALE FACTOR (METERS/INCH)
C                  RS    = SLANT RANGE VECTOR (3 VALUES)
C                  RC    = CROSS RANGE VECTOR (3 VALUES)
C
C              CONSTRUCT AIRCRAFT SILHOUETTE
C
0015          WRITE(TTYO,29)
0016          WRITE(TTYO,26)
0017          READ(TTYI,12)MODE
0018          WRITE(TTYO,17)
0019          CALL FLEASN(TTYI,AFNAME)
0020          CALL ASSIGN(AIRCFL,AFNAME,0)
0021          IF(MODE.EQ.2) GO TO 42
C
0023          WRITE(TTYO,30)
0024          CALL FLEASN(TTYI,IMFILE)
0025          DEFINE FILE 10 (512,256,U,IAV)
0026          CALL ASSIGN(NIMG,IMFILE,0)
0027          WRITE(TTYO,31)
0028          CALL FLEASN(TTYI,PARAFL)
0029          DEFINE FILE 11 (1,256,U,JAV)
0030          CALL ASSIGN(NPRM,PARAFL,0)
0031          READ(NPRM'1)NWRDS,NRECS,RMAX,SRI,CRI,
             *          (RS(I),I=1,3),(RC(I),I=1,3)
0032          GO TO 45
C
0033       40 DO 66 IZ=1,8192
```

```
0034     66 GBUF(IZ) = 0
0035        XSHFT = 0.
0036        YSHFT = 0.
0037        XLAST = -999
0038        YLAST = -999
0039        WRITE(TTYO,24)
0040        READ(TTYI,10)KTST
0041        IF(KTST.NE.NO) GO TO 42
0043        WRITE(TTYO,23)
0044        READ(TTYI,10)KTST
0045        IF(KTST.EQ.NO) GO TO 1000
C
C
C       UNTIL PROBLEM OF RE-SPECIFICATION OF RSX-11M FORTRAN IV
```



```
C       DIRECT-ACCESS FILES IS RESOLVED, DIRECT NEW FILE REQUESTS
C       WHILE IN MODE 1 TO GRACEFUL EXIT.
C
C
0047        IF(MODE.EQ.1) GO TO 999
C
C       CLEAR ALL FILE NAMES
C
0049        DO 140 IZ = 1,7
0050        AFNAME(IZ) = 0
0051        IMFILE(IZ) = 0
0052   140  PARAFL(IZ) = 0
C
C
0053        CALL CLOSE(AIRCFL)
0054        WRITE(TTYO,17)
0055        CALL FLEASN(TTYI,AFNAME)
0056        CALL ASSIGN(AIRCFL,AFNAME,0)
0057        IF(MODE.EQ.2)GO TO 41
0059        CALL CLOSE(NIMG)
0060        WRITE(TTYO,30)
0061        CALL FLEASN(TTYI,IMFILE)
0062        DEFINE FILE 10 (512,256,U,IAV)
0063        CALL ASSIGN(NIMG,IMFILE,0)
0064        CALL CLOSE(NPRM)
0065        WRITE(TTYO,31)
0066        CALL FLEASN(TTYI,PARAFL)
0067        DEFINE FILE 11 (1,256,U,JAV)
0068        CALL ASSIGN(NPRM,PARAFL,0)
0069        READ (NPPM'1)NWRDS,NRECS,RMAX,SRI,CRI,
```

```
     *              (RS(I),I=1,3),(RC(I),I=1,3)
0077  41      CONTINUE
0078  42      IF(MODE.EQ.1) GO TO 45
0079          WRITE(TTYO,18)
0080          READ(TTYI,11)SCALE
0081          WRITE(TTYO,16)
0082          READ(TTYI,10)KTST
0083          IF(KTST.EQ.NO) GO TO 42
0084      43  CALL ROTATE (RS,RC)
C     43  WRITE(TTYO,19)
C         READ(TTYI,11)(RS(I),I=1,3)
C         WRITE(TTYO,20)
C         READ(TTYI,11)(RC(I),I=1,3)
C         WRITE(TTYO,16)
C         READ(TTYI,10)KTST
C         IF(KTST.EQ.NO) GO TO 43
0085      45  CALL TVECT(RS,RC,RZ,EPS,KCMP)
0086          IF(KCMP.EQ.0) GO TO 48
0087          WRITE(TTYO,27)
0088          GO TO 43
C
```



```
0089      48  CONTINUE
0090          IF(MODE.EQ.2) GO TO 59
0091          CALL PASS2(NIMG,DMIN,DBTHR,RIMX,
     *     NWRDS,NPECS,RMAX,SRI,CRI)
C
C
C         REQUEST USER TO ENTER GRAPHIC MEMORY NUMBER (1-3):
C
C
0092      59  WRITE(TTYO,106)
0093          READ(TTYI,12) IGMN
0094          IF(IGMN.GT.0.AND.IGMN.LT.4) GO TO 49
0095          WRITE(TTYO,107) IGMN
0096          GO TO 59
C
C
0097      49  CALL DRAW(AIRCFL,RS,RC,RZ,SCALE,FINUIS,DTR,
     *              NCPI,MODE,SSZ,SDT,RIMX,ISZ,IDIF,KCMP)
C
C
C
C         BYTE-SWAP ENTIRE BUFFER GBUF, DUE TO COMTAL RSX HANDLER
C
C
0098          CALL IBYTEX(GBUF,4096)
C         NOW,WRITE GBUF TO THE COMTAL
```

```
C
C
C
0097      60 DO 700 M5=1,256
0098         I5 = (M5-1)*32+1
0099         J5 = (M5*32)
0100         DO 699 M5=I5,J5
0101         GBUF2(M5-I5+1) = GBUF(M5)
0102        699 CONTINUE
0103         CALL IWGM(IGMN,M5,GBUF2)
0104        700 CONTINUE
C
C
C
```

```
0123      12 FORMAT(I4)
0124      16 FORMAT(' ARE YOU SURE  -  ')
0125      17 FORMAT(' SILHOUETTE FILE NAME ',/)
0126      18 FORMAT(' SCALE FACTOR (METERS/INCH) =  ')
0127      19 FORMAT(' SLANT RANGE VECTOR (3 VALUES) =  ')
0128      20 FORMAT(' CROSS RANGE VECTOR (3 VALUES) =  ')
0129      23 FORMAT(',' NEW FILE (YES/NO) ')
0130      24 FORMAT(',' NEW PROJECTION (YES/NO) ')
0131      25 FORMAT(',' DISPLAY BUFFER INDEX =  ',I5,/)
0132      26 FORMAT(',' ENTER OPERATIONAL MODE :',/,
                *                1 = IMAGE DRIVEN MODE ',/,
                *                2 = STAND-ALONE MODE ')
0133      27 FORMAT(//,' ERROR  CONDITION  -  ',/,
                *               -----  RS AND RC MUST BE  ',/,
                *                      ORTHOGONAL UNIT VECTORS  ',///)
0134      28 FORMAT(//,' ERROR  CONDITION  -  ',/,
                *               -----  DISPLAY BUFFER (IBUF)  ',/,
                *                      IS ABOUT TO OVERFLOW  ',///)
0135      29 FORMAT(//,' AIRCRAFT SILHOUETTE DISPLAY',/,
                *               ---------------------------',///)
0136  30    FORMAT(',  IMAGE FILE NAME ',/)
0137  31    FORMAT(/,  PARAMETER FILE NAME ',/)
0138      32 FORMAT(2I4)
0139     101 FORMAT(' TRANSLATE GRAPHIC (YES/NO) ')
0140     102 FORMAT(' ENTER TRANSLATIONS: X,Y (REAL,SIGNED NUMBERS)')
0141     103 FORMAT(2F6.3)
0142     105 FORMAT(' PROGRAM "SILHOUETTE" TERMINATED BY OPERATOR')
0143     106 FORMAT(' ENTER COMTAL GRAPHIC MEMORY NUMBER (1-3): ')
0144     107 FORMAT(' IGMN ENTERED, ',I4,'  IS ILLEGAL. PLEASE RE-ENTER.')
0145     108 FORMAT(' PLEASE RERUN "COMTAL" IF NEW FILES TO BE SPECIFIED')
0146         END
```

```
FORTRAN IV      STORAGE MAP

NAME     OFFSET  ATTRIBUTES

GBUF2    000006  LOGICAL*1 ARRAY (32)
AFNAME   000046  INTEGER*2 ARRAY (7)
IMFILE   000064  INTEGER*2 ARRAY (7)
PARMFL   000102  INTEGER*2 ARRAY (7)
RS       000120  REAL*4    ARRAY (3)
RC       000134  REAL*4    ARRAY (3)
RZ       000150  REAL*4    ARRAY (3)
FINDIS   000164  REAL*4    ARRAY (2)
AIRCFL   000174  INTEGER*2 VARIABLE
TTYI     000202  INTEGER*2 VARIABLE
TTYO     000204  INTEGER*2 VARIABLE
NIMG     000176  INTEGER*2 VARIABLE
NPPM     000200  INTEGER*2 VARIABLE
NO       000206  INTEGER*2 VARIABLE
DTR      000210  REAL*4    VARIABLE
```

SSI	000214	REAL*4	VARIABLE
SP	000220	REAL*4	VARIABLE
EP	000224	REAL*4	VARIABLE
MODI	000230	INTEGER*2	VARIABLE
DMIN	000232	REAL*4	VARIABLE
DFTMP	000236	REAL*4	VARIABLE
MODE	002130	INTEGER*2	VARIABLE
FLE-SN	000000	REAL*4	PROCEDURE
ASSIGN	000000	REAL*4	PROCEDURE
IAD	002132	INTEGER*2	VARIABLE
JAD	002134	INTEGER*2	VARIABLE
NLRDS	002136	INTEGER*2	VARIABLE
NRETS	002140	INTEGER*2	VARIABLE
RMAX	002142	REAL*4	VARIABLE
SRI	002146	REAL*4	VARIABLE
ERI	002152	REAL*4	VARIABLE
I	002156	INTEGER*2	VARIABLE
II	002160	INTEGER*2	VARIABLE
LAST	002162	INTEGER*2	VARIABLE
CLOSE	000000	REAL*4	PROCEDURE
SCALE	002164	REAL*4	VARIABLE
RCREATE	000000	REAL*4	PROCEDURE
TRESET	000000	REAL*4	PROCEDURE
NCMP	002170	INTEGER*2	VARIABLE
RWRITE	000000	REAL*4	PROCEDURE
RTMP	002172	REAL*4	VARIABLE
ICMP	002176	INTEGER*2	VARIABLE
DFILE	000000	REAL*4	PROCEDURE
ISC	002200	INTEGER*2	VARIABLE
IDIF	002202	INTEGER*2	VARIABLE
IERROR	000000	INTEGER*2	PROCEDURE
NS	002204	INTEGER*2	VARIABLE
IS	002206	INTEGER*2	VARIABLE
JS	002210	INTEGER*2	VARIABLE
MS	002212	INTEGER*2	VARIABLE
IJGT	000000	INTEGER*2	PROCEDURE
T	002214	REAL*4	VARIABLE

FORTRAN IV STORAGE MAP

NAME	OFFSET	ATTRIBUTES
/S	002220	REAL*4 VARIABLE

COMMON BLOCK /GBUF/ LENGTH 020010

GBUF	000000	LOGICAL*1 ARRAY (8192)
G	020000	REAL*4 VARIABLE
G	020004	REAL*4 VARIABLE

```
COMMON BLOCK  LOWLFT    LENGTH 000020

XLL       000000  REAL*4      VARIABLE
YLL       000004  REAL*4      VARIABLE
XSHFT     000010  REAL*4      VARIABLE
YSHFT     000014  REAL*4      VARIABLE

COMMON BLOCK  LAST      LENGTH 000004

XLAST     000000  INTEGER*2 VARIABLE
YLAST     000002  INTEGER*2 VARIABLE

FOR\
```

```
FORTRAN IV      V01C-03         FRI 28-SEP-79 09:20:36          PAGE 001
CORE=0SK, UIC=[1,123]                                ,TT0:=DK3:SLSUBS.FTN

C     ******************************************************
C     *                                                    *
C     *                                                    *
C     *     THESE ROUTINES MODIFIED BY THE MEASUREMENT     *
C     *     CONCEPT CORP. TO SUPPORT THE COMTAL            *
C     *     VISION ONE OPERATING UNDER RSX-11M.            *
C     *                                                    *
C     *     MODS. PERFORMED TO THIS SRC SOFTWARE           *
C     *     DURING 9/79 BY ROBERT A. MCKIE, SENIOR         *
C     *     ASSOCIATE.                                     *
C     *                                                    *
C     ******************************************************
C
C
0001        SUBROUTINE DRAW (AIRCFL,RS,RC,RZ,SCALE,FINVIS,DTR,
     1                       NCRI,MODE,SSZ,SDT,RIMX,ISZ,IDIF,KCMP)
0002        COMMON /SCALE/ SCL
0003        COMMON /LAST/ XLAST,YLAST
0004        COMMON /LOWLFT/ XLL,YLL,XSHFT,YSHFT
0005        COMMON /DEBUG/ IFLG
0006        COMMON /GBUF/ GBUF,CX,CY
0007        LOGICAL*1 GBUF(8192)
0008        REAL FINVIS(2),RS(3),RC(3),RZ(3)
0009        REAL COORD(6)
0010        INTEGER AIRCFL
C
C
C         READ AIRCRAFT DESIGN FROM FILE
C         CONSTRUCT AIRCRAFT SILHOUETTE
C
C             FILE STRUCTURE
C             --------------
C
C                 4 WORD RECORDS (1 FUNCTION CODE +
C                                 3 COORDINATES)
C
C                 FUNCTION CODE DESCRIPTION
C
C                     1 - ILLUMINATE A POINT
C                     2 - ILLUMINATE A VECTOR
C                     3 - SPARE
C                     4 - ORIGINAL SCALE FACTOR
C                     5 - INITIATE A CYLINDER
C                     6 - INITIATE A CIRCLE
```

```
C                    7 = R, PHI AND THETA FOR THE CIRCLE
C                    8 = TERMINATE DISPLAY
C
C
C         READ FILE, BRANCH ON FUNCTION CODE
```



```
0011        IFLG = 0
0012        IRD = 1
0013     40 CONTINUE
0014        IF(IFLG.EQ.1) WRITE(7,808) IRD
0016    808 FORMAT(' READ LINE # ',I5)
0017        READ(AIRCFL,10)NFC,(COORD(I),I=1,3)
0018        IRD = IRD + 1
0019        GO TO (41,42,43,44,45,46,47,48,40),NFC
C
0020     41 CALL ADJUST(COORD,3)
0021        CALL DOTP(RC,COORD,X)
0022        CALL DOTP(RS,COORD,Y)
0023        CALL PLOT(X,Y,3)
0024        GO TO 50
C
0025     42 CALL ADJUST(COORD,3)
0026        CALL DOTP(RC,COORD,X)
0027        CALL DOTP(RS,COORD,Y)
0028        CALL PLOT(X,Y,2)
0029        GO TO 50
C
0030     43 GO TO 40
C
0031     44 IF(MODE.EQ.2) GO TO 31
0033        DMUAL=100.*COORD(3)*RIMX/(2.*COORD(2))
0034        GO TO 34
0035     31 IF(SCALE.NE.0) GO TO 32
0037        DMUAL=COORD(1)
0038        GO TO 34
0039     32 DMUAL=SSZ*COORD(2)*COORD(3)/(100.*SCALE)
0040     34 CALL SCAL(-DMUAL,-DMUAL,DMUAL,DMUAL)
0041        CALL PLOT(0.,0.,1)
0042        GO TO 50
C
0043     45 NCIRC=COORD(1)
0044        CALL CYLNDP(AIRCFL,RS,RC,RZ,DTR,NCRI,NCIRC)
0045        GO TO 50
C
0046     46 CONTINUE
0047        READ(AIRCFL,10)NFC,(COORD(I),I=4,6)
```

```
0048          CALL ADJUST(COORD,4)
0049          CALL CIRCLE(COORD,RS,RC,DTR,NCRI)
0050          GO TO 50
C
0051       47 GO TO 40
C
0052       48 CONTINUE
0053          GO TO 60
C
C
0054       50 GO TO 40
C
C
```



```
C          REWIND AND RETURN
C
0055       60 CONTINUE
0056          REWIND AIRCFL
0057          RETURN
C
C
0058       10 FORMAT(I6,3F6.0)
0059          END
```

FORTRAN IV STORAGE MAP

NAME	OFFSET	ATTRIBUTES
FINDIS	000026	REAL*4 PARAMETER ARRAY (2)
PE	000016	REAL*4 PARAMETER ARRAY (3)
PC	000020	REAL*4 PARAMETER ARRAY (3)
PZ	000022	REAL*4 PARAMETER ARRAY (3)
COORD	000052	REAL*4 ARRAY (6)
AIRCFL	000014	INTEGER*2 PARAMETER VARIABLE
SCALE	000024	REAL*4 PARAMETER VARIABLE
DTR	000030	REAL*4 PARAMETER VARIABLE
NCRI	000032	INTEGER*2 PARAMETER VARIABLE
MODE	000034	INTEGER*2 PARAMETER VARIABLE
SSZ	000036	REAL*4 PARAMETER VARIABLE
SDT	000040	REAL*4 PARAMETER VARIABLE

```
RIM      000042  REAL*4     PARAMETER VARIABLE
I2       000044  INTEGER*2  PARAMETER VARIABLE
I1I2     000046  INTEGER*2  PARAMETER VARIABLE
KCHF     000050  INTEGER*2  PARAMETER VARIABLE
IFD      000152  INTEGER*2  VARIABLE
MFC      000154  INTEGER*2  VARIABLE
I        000156  INTEGER*2  VARIABLE
ADJUST   000000  REAL*4     PROCEDURE
DOTP     000000  REAL*4     PROCEDURE
X        000160  REAL*4     VARIABLE
Y        000164  REAL*4     VARIABLE
PLOT     000000  REAL*4     PROCEDURE
DMVAL    000170  REAL*4     VARIABLE
SCAL     000000  REAL*4     PROCEDURE
NCIRC    000174  INTEGER*2  VARIABLE
CYLNDR   000000  REAL*4     PROCEDURE
CIRCLE   000000  REAL*4     PROCEDURE

COMMON BLOCK  SCALE'     LENGTH 000004

SCL      000000  REAL*4     VARIABLE

COMMON BLOCK  LAST       LENGTH 000010

XLAST    000000  REAL*4     VARIABLE
YLAST    000004  REAL*4     VARIABLE

COMMON BLOCK 'LOWLFT'    LENGTH 000020

XLL      000000  REAL*4     VARIABLE
YLL      000004  REAL*4     VARIABLE
XSHFT    000010  REAL*4     VARIABLE
YSHFT    000014  REAL*4     VARIABLE

COMMON BLOCK /DEBUG/     LENGTH 000002

IFLG     000000  INTEGER*2  VARIABLE

COMMON BLOCK /GBUF/      LENGTH 020010
```



```
FORTRAN IV       STORAGE MAP

NAME     OFFSET  ATTRIBUTES

GBUF     000000  LOGICAL*1 ARRAY (8192)
GX       020000  REAL*4     VARIABLE
GY       020004  REAL*4     VARIABLE
```

```
FORTRAN IV      V01C-03         FRI 28-SEP-79 09:21:15          PAGE 001
CORE=0SK, UIC=[1,123]                                ,TT0:=DK3:SLSUBS.FTN

0001          SUBROUTINE CIRCLE (COORD,RS,RC,DTR,NCRI)
0002          REAL COORD(6),RS(3),RC(3),TEMP(3)
0003          EQUIVALENCE (X,TEMP(1)),(Y,TEMP(2)),(Z,TEMP(3))
          C
          C
          C          PROJECT A CIRCLE ;
          C              CENTER (X0,Y0,Z0)       = COORD(1,2 AND 3)
          C              RADIUS (R)              = COORD(4)
          C              ORIENTATION (PHI,THETA) = COORD(5 AND 6)
          C          ONTO THE PLANE DEFINED BY THE
          C          SLANT AND CROSS RANGE VECTORS (RS,RC)
          C
          C
          C          GET PARAMETERS
          C          POSITION BEAM TO 1ST POINT
          C          INITIALIZE SINE AND COSINE FUNCTIONS
          C
0004          X0=COORD(1)
0005          Y0=COORD(2)
0006          Z0=COORD(3)
0007          R=COORD(4)
0008          PHI=COORD(5)*DTR
0009          THETA=COORD(6)*DTR
          C
0010          SPHI=SIN(PHI)
0011          CPHI=COS(PHI)
0012          STHETA=SIN(THETA)
0013          CTHETA=COS(THETA)
0014          X=R*CPHI+X0
0015          Y=R*STHETA*SPHI+Y0
0016          Z=-R*CTHETA*SPHI+Z0
0017          CALL DOTP(RC,TEMP,XP)
0018          CALL DOTP(RS,TEMP,YP)
0019          CALL PLOT(XP,YP,3)
          C
0020          SANG=SIN(DTR*360./NCRI)
0021          CANG=COS(DTR*360./NCRI)
0022          SNANG=0.
0023          CNANG=1.
          C
          C
          C          DRAW 'NCRI' VECTORS AROUND THE CIRCLE
          C
0024          N=NCRI+1
0025          DO 44 I=1,N
0026          XP=R*CNANG
0027          YP=R*SNANG
0028          X=XP*CPHI+X0
0029          Y=XP*STHETA*SPHI+YP*CTHETA+Y0
0030          Z=-XP*CTHETA*SPHI+YP*STHETA+Z0
```

```
0031          CALL DOTP(RC,TEMP,XP)
0032          CALL DOTP(RS,TEMP,YP)
0033          CALL PLOT(XP,YP,2)
0034          XP=SNANG
```

```
FORTRAN IV      V01C-03         FRI 28-SEP-79 09:21:15          PAGE 002
CORE=OSA, UIC=[1,123]                                  ,TT0:=DK3:SLSUBS.FTN

0035          YP=CNANG
0036          SNANG=XP*CANG+YP*SANG
0037       44 CNANG=YP*CANG-XP*SANG
      C
      C
0038          RETURN
0039          END
```



```
FORTRAN IV      STORAGE MAP

NAME     OFFSET  ATTRIBUTES

COORD    000014  REAL*4    PARAMETER ARRAY (6)
RS       000016  REAL*4    PARAMETER ARRAY (3)
RC       000020  REAL*4    PARAMETER ARRAY (3)
TEMP     000026  REAL*4    ARRAY (3)
DTP      000022  REAL*4    PARAMETER VARIABLE
NCRI     000024  INTEGER*2 PARAMETER VARIABLE
X        000026  REAL*4    VARIABLE
Y        000032  REAL*4    VARIABLE
Z        000036  REAL*4    VARIABLE
X0       000046  REAL*4    VARIABLE
Y0       000052  REAL*4    VARIABLE
Z0       000056  REAL*4    VARIABLE
R        000062  REAL*4    VARIABLE
PHI      000066  REAL*4    VARIABLE
THETA    000072  REAL*4    VARIABLE
SPHI     000076  REAL*4    VARIABLE
SIN      000000  REAL*4    PROCEDURE
CPHI     000102  REAL*4    VARIABLE
COS      000000  REAL*4    PROCEDURE
STHETA   000106  REAL*4    VARIABLE
CTHETA   000112  REAL*4    VARIABLE
DOTP     000000  REAL*4    PROCEDURE
XP       000116  REAL*4    VARIABLE
YP       000122  REAL*4    VARIABLE
```

```
PLOT    000000  REAL*4    PROCEDURE
S       000126  REAL*4    VARIABLE
C       000132  REAL*4    VARIABLE
SINRC   000136  REAL*4    VARIABLE
COSRC   000142  REAL*4    VARIABLE
N       000146  INTEGER*2 VARIABLE
I       000150  INTEGER*2 VARIABLE
```

```
FORTRAN IV      V01C-03         FRI 28-SEP-79 09:21:49          PAGE 001
CORE=0SK, UIC=[1,123]                                  ,TT0:=DK3:SLSUBS.FTN

0001          SUBROUTINE CYLNDR (AIRCFL,RS,RC,RZ,DTR,NCRI,NCIRC)
0002          INTEGER AIRCFL
0003          REAL COORD(6),RS(3),RC(3),RZ(3)
0004          REAL ZC(3),S(3),T(3),X(2,10),Y(2,10)
        C
        C
        C           PROJECT A CYLINDER ;
        C               DEFINED AS A SET OF CONNECTED CIRCLES
        C               PROJECT THE CIRCLES AS USUAL
        C               CONSTRUCT THE LINE SEGMENTS TO BE CONNECTED
        C               DRAW THE LINE SEGMENTS
        C
0005          NCR=MIN0(NCIRC,10)
0006          NDCR=0
        C
0007          DO 48 N=1,NCR
0008          READ(AIRCFL,10)NFC,(COORD(I),I=1,3)
0009          READ(AIRCFL,10)NFC,(COORD(I),I=4,6)
0010          CALL ADJUST(COORD,4)
0011          CALL CIRCLE(COORD,RS,RC,DTR,NCRI)
0012          U=COORD(5)*DTR
0013          ZC(1)=SIN(U)
0014          CP=COS(U)
0015          U=COORD(6)*DTR
0016          ZC(2)=-SIN(U)*CP
0017          ZC(3)=COS(U)*CP
0018          CALL CROSSP(RZ,ZC,T)
0019          CALL UNITV(T,T,U)
0020          IF(U.GE..00001) GO TO 42
0022          NDCR=NDCR+1
0023          GO TO 48
0024       42 DO 44 K=1,2
0025          L=2*K-3
0026          DO 43 I=1,3
0027       43 S(I)=T(I)*COORD(4)*L
0028          CALL DOTP(S,RC,X(K,N))
0029          CALL DOTP(COORD,RC,U)
0030          X(K,N)=X(K,N)+U
0031          CALL DOTP(S,RS,Y(K,N))
0032          CALL DOTP(COORD,RS,U)
```

```
0033      44 Y(K,N)=Y(K,N)+U
0034      48 CONTINUE
C
0035         NCR=NCR-NDCR
0036         IF(NCR.LT.2) RETURN
0037         DO 54 K=1,2
0038         CALL PLOT(X(K,1),Y(K,1),3)
0040         DO 54 N=2,NCR
0041      54 CALL PLOT(X(K,N),Y(K,N),2)
C
0042         RETURN
0043      10 FORMAT(I6,3F6.0)
0044         END
```

```
FORTRAN IV      STORAGE MAP

NAME     OFFSET  ATTRIBUTES

COORD    000032  REAL*4     ARRAY (6)
RS       000016  REAL*4     PARAMETER ARRAY (3)
RC       000020  REAL*4     PARAMETER ARRAY (3)
RI       000022  REAL*4     PARAMETER ARRAY (3)
C        000062  REAL*4     ARRAY (3)
S        000076  REAL*4     ARRAY (3)
T        000112  REAL*4     ARRAY (3)
X        000126  REAL*4     ARRAY (2,10) VECTORED
Y        000246  REAL*4     ARRAY (2,10) VECTORED
RIFIFL   000014  INTEGER*2  PARAMETER VARIABLE
DTR      000024  REAL*4     PARAMETER VARIABLE
NCPI     000026  INTEGER*2  PARAMETER VARIABLE
NCIPC    000030  INTEGER*2  PARAMETER VARIABLE
NCP      000436  INTEGER*2  VARIABLE
MINO     000000  INTEGER*2  PROCEDURE
NDCP     000440  INTEGER*2  VARIABLE
N        000442  INTEGER*2  VARIABLE
NFC      000444  INTEGER*2  VARIABLE
I        000446  INTEGER*2  VARIABLE
ADJUST   000000  REAL*4     PROCEDURE
CIRCLE   000000  REAL*4     PROCEDURE
U        000450  REAL*4     VARIABLE
SIN      000000  REAL*4     PROCEDURE
CE       000454  REAL*4     VARIABLE
COS      000000  REAL*4     PROCEDURE
CROSSP   000000  REAL*4     PROCEDURE
UNITV    000000  REAL*4     PROCEDURE
K        000460  INTEGER*2  VARIABLE
L        000462  INTEGER*2  VARIABLE
DOTP     000000  REAL*4     PROCEDURE
PLOT     000000  REAL*4     PROCEDURE
```

```
FORTRAN IV      V01C-03         FRI 28-SEP-79 09:22:07          PAGE 001
CORE-05A, UIC-[1,123]                                ,TT0:=DK3:SLSUBS.FTN

0001          SUBROUTINE PASS2 (NIMG,DMIN,DBTHR,RIMX,
     *                          NURDS,NRECS,RMAX,SRI,CRI)
0002          REAL D(256)
0003          LOGICAL*1 IMLINE(256)
0004          LOGICAL*1 ITEXT(8)
0005          INTEGER TTYO,TTYI
0006          DATA TTYO/7/,TTYI/5/
0007          DATA ITEXT/'C','L',' ','I',' ','1',' ','8'/
C
C
C             DISPLAY RADAR IMAGE RETURN
C             --------------------------
C
C                 DETERMINE SLANT AND CROSS RANGE SCALE FACTORS
C                 READ IMAGE RECORDS
C                 CONVERT TO DB
C                 CALCULATE INTENSITY (1-256)
C                 DISPLAY POINTS
C
0008          CRD=NURDS*CRI
0009          SRD=NRECS*SRI
0010          IF(CRD.GT.SRD) GO TO 32
0012          SRMAX=NRECS
0013          CRMAX=NURDS*SRD/CRD
0014          RIMX=SRD
0015          CR1=.5*(NURDS-CRMAX)
0016          CR2=CRMAX+CR1
0017          XSP = (CR2-CR1)
0018          YSP = SRMAX
0019          XINC = 256./XSP
0020          YINC = 256./YSP
0021          XMIN = CR1
0022          YMIN = 0.
0023          GO TO 40
0024       32 CRMAX=NURDS
0025          SRMAX=NRECS*CRD/SRD
0026          RIMX=CRD
0027          SR1=.5*(NRECS-SRMAX)
0028          SR2=SRMAX+SR1
0029          XSP = CRMAX
0030          YSP = SR2-SR1
0031          XINC = 256./XSP
0032          YINC = 256./YSP
0033          XMIN = 0.
0034          YMIN = SR1
C
0035       40 WRITE(TTYO,100)
0036          READ(TTYI,101) IMNO
```

```
0037          IF(IMNO.GT.0.AND.IMNO.LT.4) GO TO 75
0039          WRITE(TTYO,102) IMNO
0040          GO TO 40
```



```
C     CLEAR IMAGE BUFFER
C
0041     75 IF(IMNO.EQ.1) ITEXT(6) = 49
0043        IF(IMNO.EQ.2) ITEXT(6) = 50
0045        IF(IMNO.EQ.3) ITEXT(6) = 51
0047        CALL IWCB(ITEXT)
C
C
0048        DO 48 I=1,NRECS
0049        DO 60 I8 = 1,256
0050        D(I8) = 0.
0051     60 IMLINE(I8) = 0
0052        READD(NIMG'I)(D(K),K=1,NWRDS)
0053        DMAX=DMIN
0054        DO 42 K=1,NWRDS
0055        RN=D(K)/RMAX
0056        D(K)=DMIN
0057        IF(RN.LE.0) GO TO 42
0058        D(K)=AMAX1(DMIN,20.*ALOG10(RN))
0060        DMAX=AMAX1(D(K),DMAX)
0061     42 CONTINUE
0062        IF(DMAX.LE.DBTHR) GO TO 48
0063        DO 44 K=1,NWRDS
0065        X=(DBTHR-D(K))/DBTHR
0066        INTNS=AMAX1(0.,X)*256.
C
C
0067        KKK = FLOAT(K)*XINC
0068        IF(KKK.LT.1.OR.KKK.GT.256) GO TO 44
0070        IMLINE(KKK) = INTNS
0071     44 CONTINUE
C
C
C     NOW, WRITE A LINE TO THE COMTAL
C
C     INVERT IMAGE Y; SCALE Y
C
0072        LLL = (257.-(FLOAT(I)*YINC))
0073        IF(LLL.LT.1.OR.LLL.GT.256) GO TO 48
0075        CALL IWIM(IMNO,LLL,IMLINE)
0076     48 CONTINUE
C
```

```
0077      50 RETURN
0078     100 FORMAT(' ENTER COMTAL IMAGE MEMORY NUMBER: (1-3) ')
0079     101 FORMAT(I4)
0080     102 FORMAT(' IMNO = ',I4,'  ILLEGAL. PLEASE RE-ENTER.')
0081     150 FORMAT(A2)
0082         END
```

```
FORTRAN IV      STORAGE MAP

NAME    OFFSET  ATTRIBUTES

D       000036  REAL*4    ARRAY (256)
IMLINE  002036  LOGICAL*1 ARRAY (256)
ITEXT   002436  LOGICAL*1 ARRAY (8)
NIMG    000014  INTEGER*2 PARAMETER VARIABLE
DMIN    000016  REAL*4    PARAMETER VARIABLE
DETHR   000020  REAL*4    PARAMETER VARIABLE
RIMG    000022  REAL*4    PARAMETER VARIABLE
NWRDS   000024  INTEGER*2 PARAMETER VARIABLE
NREIS   000026  INTEGER*2 PARAMETER VARIABLE
RMAX    000030  REAL*4    PARAMETER VARIABLE
SRI     000032  REAL*4    PARAMETER VARIABLE
CRI     000034  REAL*4    PARAMETER VARIABLE
TTY0    002446  INTEGER*2 VARIABLE
TTY1    002450  INTEGER*2 VARIABLE
CR0     002616  REAL*4    VARIABLE
SR0     002622  REAL*4    VARIABLE
SRMAX   002626  REAL*4    VARIABLE
CRMAX   002632  REAL*4    VARIABLE
CR1     002636  REAL*4    VARIABLE
CR2     002642  REAL*4    VARIABLE
XSP     002646  REAL*4    VARIABLE
YSP     002652  REAL*4    VARIABLE
XINC    002656  REAL*4    VARIABLE
YINC    002662  REAL*4    VARIABLE
XMIN    002666  REAL*4    VARIABLE
YMIN    002672  REAL*4    VARIABLE
SR1     002676  REAL*4    VARIABLE
SR2     002702  REAL*4    VARIABLE
IMNO    002706  INTEGER*2 VARIABLE
INCR    000000  INTEGER*2 PROCEDURE
I       002710  INTEGER*2 VARIABLE
I3      002712  INTEGER*2 VARIABLE
K       002714  INTEGER*2 VARIABLE
DMAX    002716  REAL*4    VARIABLE
RN      002722  REAL*4    VARIABLE
AMAX1   000000  REAL*4    PROCEDURE
ALOG10  000000  REAL*4    PROCEDURE
X       002726  REAL*4    VARIABLE
INTNS   002732  INTEGER*2 VARIABLE
KKK     002734  INTEGER*2 VARIABLE
```

```
FLOAT   000000  REAL*4     PROCEDURE
LL=     002736  INTEGER*2  VARIABLE
IL:I    000000  INTEGER*2  PROCEDURE
```

```
FORTRAN IV      V01C-03          FRI 28-SEP-79 09:22:51          PAGE 001
CORE=08K, UIC=[1,123]                                ,TT0:=DK3:SLSUBS.FTN

0001          SUBROUTINE TVECT (RS,RC,RZ,EPS,KCMP)
0002          REAL RS(3),RC(3),RZ(3)
        C
        C
        C           COMPUTE RZ VECTOR
        C           CHECK RS AND RC VECTORS
        C           RS AND RC MUST BE ORTHOGONAL UNIT VECTORS
        C
        C
0003          KCMP=0
0004          CALL CROSSP(RS,RC,RZ)
        C
0005          X=SQRT(RS(1)**2+RS(2)**2+RS(3)**2)
0006          IF(ABS(X-1.).LT.EPS) GO TO 44
0008          KCMP=1
0009          GO TO 50
        C
0010       44 X=SQRT(RC(1)**2+RC(2)**2+RC(3)**2)
0011          IF(ABS(X-1.).LT.EPS) GO TO 45
0013          KCMP=1
0014          GO TO 50
        C
0015       45 CALL DOTP(RS,RC,X)
0016          IF(ABS(X).LT.EPS) GO TO 50
0018          KCMP=1
        C
0019       50 RETURN
0020          END
```

```
FORTRAN IV      STORAGE MAP

NAME    OFFSET  ATTRIBUTES

PS      000014  REAL*4     PARAMETER ARRAY (3)
RC      000016  REAL*4     PARAMETER ARRAY (3)
RZ      000020  REAL*4     PARAMETER ARRAY (3)
EPS     000022  REAL*4     PARAMETER VARIABLE
KCMP    000024  INTEGER*2  PARAMETER VARIABLE
```

```
CROSSP  000000  REAL*4     PROCEDURE
A       000026  REAL*4     VARIABLE
SVET    000000  REAL*4     PROCEDURE
WES     000000  REAL*4     PROCEDURE
DOTP    000000  REAL*4     PROCEDURE
```

```
FORTRAN IV      V01C-03         FRI 28-SEP-79 09:23:04          PAGE 001
CORE=08K, UIC=[1,123]                                  ,TT0:=DK3:SLSUBS.FTN

0001          SUBROUTINE PLOT (X,Y,N)
0002          COMMON /GBUF/ GBUF,CX,CY
0003          LOGICAL*1 GBUF(8192)
          C
          C
          C         ILLUMINATE ABSOLUTE POINT COORDINATE
          C         FROM PREVIOUS BEAM POSITION
          C
          C
0004          GO TO (48,43,44),N
          C
0005      43  CALL LVECT(X-CX,Y-CY)
0006          GO TO 48
          C
0007      44  CALL RDOT(X-CX,Y-CY)
          C
0008      48  CX=X
0009          CY=Y
          C
0010      50  RETURN
0011          END
```

```
FORTRAN IV      STORAGE MAP

NAME    OFFSET  ATTRIBUTES

X       000014  REAL*4     PARAMETER VARIABLE
Y       000016  REAL*4     PARAMETER VARIABLE
N       000020  INTEGER*2  PARAMETER VARIABLE
LVECT   000000  INTEGER*2  PROCEDURE
RDOT    000000  REAL*4     PROCEDURE

COMMON BLOCK /GBUF/     LENGTH 020010
```

```
GBUF    000000  LOGICAL*1 ARRAY (8192)
C       020000  REAL*4    VARIABLE
C       020004  REAL*4    VARIABLE



FORTRAN IV      V01C-03         FRI 28-SEP-79 09:23:17          PAGE 001
CORE=08K, UIC=[1,123]                                  ,TT0:=DK3:SLSUBS.FTN

      C
      C
      C
      C
      C
      C
0001          SUBROUTINE FLEASN(LUN,FILNAM)
0002          INTEGER*2 FILNAM(7)
0003          READ(LUN,5)(FILNAM(I),I=1,7)
0004        5 FORMAT(7A2)
0005          RETURN
0006          END
```



```
FORTRAN IV      STORAGE MAP

NAME    OFFSET  ATTRIBUTES

FILNAM  000016  INTEGER*2 PARAMETER ARRAY (7)
LUN     000014  INTEGER*2 PARAMETER VARIABLE
I       000024  INTEGER*2 VARIABLE

FOR,
```

```
C     ************************************************************
C     *                                                          *
C     *     THESE ROUTINES ARE A SUBSET OF THE                   *
C     *     VT-11 FORTRAN SUPPORT ROUTINES CALLED                *
C     *     BY THE SRC 'SILHOUETTE' PROGRAM, RE-WRITTEN          *
C     *     FOR THE COMTAL VISION ONE.                           *
C     *                                                          *
C     *     WRITTEN BY: ROBERT A MCKIE,SENIOR ASSOCIATE          *
C     *                 MEASUREMENT CONCEPT CORP.                *
C     *                       9/79                               *
C     *                                                          *
C     ************************************************************
C
C
      SUBROUTINE SCAL(X1,Y1,X2,Y2)
C
      COMMON /SCALE/ SCL
      COMMON /LOWLFT/XLL,YLL,XSHFT,YSHFT
C     SUBROUTINE TO REPLACE VT-11 SCAL ROUTINE
C     TO COMPUTE A SCALE FACTOR FOR SUBSEQUENT DRAWINGS
C     ON THE COMTAL VISION ONE. UNIFORM SCALING IN BOTH X
C     AND Y DIRECTIONS IS COMPUTED, BASED ON WHETHER
C     THE DISTANCE BETWEEN X1 AND X2 IS SMALLER OR GREATER THAN THAT
C     BETWEEN Y1 AND Y2: THE GREATER DISTANCE IS USED TO DETERMINE
C     SCALE 'SCL'.
C
      A=X2 - X1
      B=Y2 - Y1
      XLL = 512. + (4.* XSHFT)
      YLL = 512. + (4. * YSHFT)
      IF(ABS(A).LT.ABS(B)) GO TO 10
      SCL = 1024./A
      GO TO 20
   10 SCL = 1024./B
   20 CONTINUE
      RETURN
      END
```

```
FORTRAN IV      STORAGE MAP

NAME    OFFSET  ATTRIBUTES

X1      000014  REAL*4    PARAMETER VARIABLE
Y1      000016  REAL*4    PARAMETER VARIABLE
X2      000020  REAL*4    PARAMETER VARIABLE
Y2      000022  REAL*4    PARAMETER VARIABLE
A       000024  REAL*4    VARIABLE
B       000030  REAL*4    VARIABLE
ABS     000000  REAL*4    PROCEDURE

COMMON BLOCK /SCALE/    LENGTH 000004

SCL     000000  REAL*4    VARIABLE

COMMON BLOCK /LOWLFT    LENGTH 000020

XLL     000000  REAL*4    VARIABLE
YLL     000004  REAL*4    VARIABLE
XSHFT   000010  REAL*4    VARIABLE
YSHFT   000014  REAL*4    VARIABLE
```



```
FORTRAN IV      V01C-03         FRI 28-SEP-79 09:25:01          PAGE 001
/CORE:08K, UIC:[1,123]                                  ,TT0:=DK3:COMGRF.FTN

      C
      C
      C
      C
      C
0001        SUBROUTINE APNT(X,Y,IP1,IP2,IP3,IP4)
      C
      C
0002        CX = X
0003        CY = Y
      C
      C
0004        RETURN
0005        END
```

```
FORTRAN IV      STORAGE MAP

NAME    OFFSET  ATTRIBUTES

X       000014  REAL*4     PARAMETER VARIABLE
Y       000016  REAL*4     PARAMETER VARIABLE
IP1     000020  INTEGER*2  PARAMETER VARIABLE
IP2     000022  INTEGER*2  PARAMETER VARIABLE
IP3     000024  INTEGER*2  PARAMETER VARIABLE
IP4     000026  INTEGER*2  PARAMETER VARIABLE
CX      000030  REAL*4     VARIABLE
CY      000034  REAL*4     VARIABLE
```

```
FORTRAN IV      V01C-03         FRI 28-SEP-79 09:25:11          PAGE 001
CORE=08K, UIC=[1,123]                                  ,TT0:=DK3:COMGRF.FTN

      C
      C
      C
      C
      C
0001        SUBROUTINE RDOT(DX,DY)
      C
      C
0002        COMMON /GBUF/ GBUF,CX,CY
0003        LOGICAL*1 GBUF(8192)
      C
      C
0004        IX=CX+DX
0005        IY=CY+DY
0006        CX=CX+DX
0007        CY=CY+DY
      C
      C
0008        CALL CDOT (IX,IY)
0009        RETURN
0010        END
```

```
FORTRAN IV      STORAGE MAP

NAME    OFFSET  ATTRIBUTES

DX      000014  REAL*4     PARAMETER VARIABLE
DY      000016  REAL*4     PARAMETER VARIABLE
```

```
I%      000020  INTEGER*2 VARIABLE
I%      000022  INTEGER*2 VARIABLE
CDOT    000000  REAL*4    PROCEDURE

COMMON BLOCK  GBUF/     LENGTH 020010

GBUF    000000  LOGICAL*1 ARRAY (8192)
CX      020000  REAL*4    VARIABLE
CY      020004  REAL*4    VARIABLE



FORTRAN IV      V01C-03         FRI 28-SEP-79 09:25:22          PAGE 001
CORE=08K, UIC=[1,123]                                   ,TT0:=DK3:COMGRF.FTN

                C
                C
                C
0001                SUBROUTINE LUECT(DX,DY)
                C
                C       SIMULATES VT-11 LUEC, WHICH DRAWS A CONTINUOUS LINE BETWEEN
                C       CURRENT CURSOR LOCATION, (X,Y) , AND (X+DX,Y+DY). AS THE COMTAL GRAPHIC
                C       MEMORY IS A 256 X 256 DOT MATRIX, THIS ROUTINE WILL GENERATE
                C       THE NECESSARY DOTS TO APPROXIMATE THE INDICATED STRAIGHT LINE.
                C       PARAMETER 'T' IS USED TO ROUTE LUECT CALLS OF NEAR ZERO OR INFINITE
                C       SLOPE TO STMT # 5,(Y=F(X)), OR TO STMT # 55 (X=G(Y)), OR TO RETURN
                C       WITHOUT DRAWING IF BOTH Dx AND DY ARE VERY SMALL.
                C
                C
0002                COMMON /GBUF/ GBUF,CX,CY
0003                COMMON /DEBUG/ IFLG
0004                LOGICAL*1 GBUF(8192)
0005                REAL M,B,T,MP,BP
0006                INTEGER Y1,Y2,X1,X2
0007                DATA T/.000005/
                C
                C       COMPUTE: EQUATION OF LINE THROUGH (CX,CY) AND (CX+DX,CY+DY)
                C
                C       CHECK FOR INFINITE SLOPE!
                C
0008                IF(ABS(DX).LT.T.AND.ABS(DY).LT.T) GO TO 11
0010                IF(ABS(DX).LT.T) GO TO 55
0012                IF(ABS(DY).LT.T) GO TO 5
0014                IF(ABS(DY).GT.ABS(DX)) GO TO 55
                C
                C
```

```
C
C     THIS CODE FOR Y = F(X)
C
C
0016      5 M = DY/DX
0017        B = CY - M*CX
C
C
0018        DO 10  I=1,IABS(INT(DX))
0019        X = CX + SIGN(FLOAT(I),DX)
C
C       COMPUTE: 'TRUE (FLOATING POINT) Y'
C
0020        TY = M*X + B
C
C       DECIDE WHICH IS 'NEARER' PIXEL
```



```
C
0021        YI = INT(TY)
0022        Y2 = YI + ISIGN(1,INT(DY))
C
0023        IF(ABS(TY-FLOAT(Y1)).GT.ABS(TY-FLOAT(Y2))) GO TO 30
C
C
C
0025     20 IY=Y1
0026        GO TO 50
C
C
0027     30  IY = Y2
C
C       THE CALL TO CDOT WILL PLACE AN ILLUMINATED POINT AT POSITION (X,Y)
C       OF THE 256 X 256 COMTAL GRAPHIC MEMORY.
C
C
C
0028     50 IX=X
0029        CALL CDOT(IX,IY)
C
0030     10 CONTINUE
0031     11 CX=CX+DX
0032        CY=CY+DY
0033        RETURN
C
C
C
C       THIS CODE FOR X = F(Y)
```

```
C
C
   55 MP = DX/DY
      BP = CX-MP*CY
      DO 60  I=1,IABS(INT(DY))
      Y=CY + SIGN(FLOAT(I),DY)
      TX = MP*Y + BP
      X1 = INT(TX)
      X2 = X1 + ISIGN(1,INT(DX))
      IF(ABS(TX-FLOAT(X1)).GT.ABS(TX-FLOAT(X2))) GO TO 70
      IX =X1
      GO TO 75
   70 IX = X2
   75 IV = Y
C
C     CALL CDOT TO ILLUMINATE PIXEL
C
C
      CALL CDOT(IX,IY)
   60 CONTINUE
      CX = CX   + DX
      CY = CY + DY
      RETURN
```



```
      END
```

FORTRAN IV STORAGE MAP

NAME	OFFSET	ATTRIBUTES
DX	000014	REAL*4 PARAMETER VARIABLE
DY	000016	REAL*4 PARAMETER VARIABLE
M	000026	REAL*4 VARIABLE
B	000032	REAL*4 VARIABLE
T	000020	REAL*4 VARIABLE
MP	000036	REAL*4 VARIABLE
BP	000042	REAL*4 VARIABLE
Y1	000046	INTEGER*2 VARIABLE
Y2	000050	INTEGER*2 VARIABLE
X1	000052	INTEGER*2 VARIABLE

```
X2          000054  INTEGER*2 VARIABLE
ABS         000000  REAL*4    PROCEDURE
I           000056  INTEGER*2 VARIABLE
IABS        000000  INTEGER*2 PROCEDURE
INT         000000  INTEGER*2 PROCEDURE
X           000060  REAL*4    VARIABLE
SIGN        000000  REAL*4    PROCEDURE
FLOAT       000000  REAL*4    PROCEDURE
TY          000064  REAL*4    VARIABLE
ISIGN       000000  INTEGER*2 PROCEDURE
IY          000070  INTEGER*2 VARIABLE
IX          000072  INTEGER*2 VARIABLE
CDOT        000000  REAL*4    PROCEDURE
Y           000074  REAL*4    VARIABLE
TX          000100  REAL*4    VARIABLE

COMMON BLOCK /GBUF/     LENGTH 020010

GBUF        000000  LOGICAL*1 ARRAY (8192)
CX          020000  REAL*4    VARIABLE
CY          020004  REAL*4    VARIABLE

COMMON BLOCK /DEBUG/    LENGTH 000002

IFLG        000000  INTEGER*2 VARIABLE
```



```
C
C
C
0001          SUBROUTINE CDOT(X,Y)
C
C             CDOT SETS A BIT IN ARRAY 'GBUF' CORRESPONDING TO PIXEL
C             LOCATION (X,Y). GBUF IS ORGANIZED AS CONSEQUTIVE
C             'LINES' OF BITS, WHERE 256 BITS = 32 BYTES.  THE LINES
C             ARE STORED IN DESCENDING ORDER (256-1), SO THAT THE LOWER LEFT
C             ORIGIN CONVENTION OF THE VT-11 IS PRESERVED.
C
C
C
0002          COMMON /GBUF/ GBUF,CX,CY
0003          COMMON /LOWLFT/ XLL,YLL,XSHFT,YSHFT
0004          COMMON /DEBUG/ IFLG
0005          COMMON /LAST/ XLAST,YLAST
0006          LOGICAL*1 GBUF(8192)
0007          LOGICAL*1 BITS(8)
```

```
0005          INTEGER*2 WINDEX,BINDEX,X,Y,XLAST,YLAST
0006          DATA BITS/128,64,32,16,8,4,2,1/
C
C             CONVERT TO COMTAL COORDINATE FRAME BY TRANSLATING LOWER LEFT CORNER
C              TO (0,0) AND REDUCING RASTER COUNT BY A FACTOR OF 4 (1024/4 = 256).
C
0010          XT = (FLOAT(X)+XLL)/4.
0011          YT = (FLOAT(Y)+YLL)/4.
C
0012          X = (X+INT(XLL))/4
0013          Y = (Y+INT(YLL))/4
C
C
C             DETERMINE 'NEAREST' PIXEL
C
0014          IF((XT-FLOAT(X)).GT.(.5)) X= X+1
0016          IF((YT-FLOAT(Y)).GT.(.5)) Y = Y+1
C
C             NOW IN COMTAL FRAME
C
C             CHECK TO SEE IF SAME AS LAST POINT PLOTTED
C             THIS SAVES A LITTLE TIME BY NOT REPEATING
C             THE STORAGE OF SEQUENTIALLY IDENTICAL X,Y'S.
C
0018          IF(Y.EQ.XLAST.AND.Y.EQ.YLAST) GO TO 35
C
C             NOT SAME, SO RESET XLAST, YLAST
```



```
C
0020          XLAST = X
0021          YLAST = Y
C
C             NOW, CHECK RANGE OF X AND Y. ARE THEY WITHIN 0-255?
C
0022          IF(X.LT.0.OR.X.GT.255) GO TO 35
0024          IF(Y.LT.0.OR.Y.GT.255) GO TO 35
C
C             OK - PROCEED
C
C             COMPUTE: INDEX TO GBUF CORRESPONDING TO X,Y
C
```

```
C     DEVELOP WORD INDEX
      5 UINDEX = ((256-V)-1)*32 + (X/8) + 1

C     DEVELOP BIT INDEX

      BINDEX = MOD((X+1),8)
      IF(BINDEX.EQ.0) BINDEX = 8

      J=GBUF(UINDEX)
      K=BITS(BINDEX)
      GBUF(UINDEX) = IOR(J,K)
   35 RETURN
      END
```



```
FORTRAN IV      STORAGE MAP

NAME    OFFSET  ATTRIBUTES

BITS    000020  LOGICAL*1 ARRAY (8)
X       000014  INTEGER*2 PARAMETER VARIABLE
V       000016  INTEGER*2 PARAMETER VARIABLE
UINDEX  000032  INTEGER*2 VARIABLE
BINDEX  000034  INTEGER*2 VARIABLE
Z*      000036  REAL*4    VARIABLE
FLOAT   000000  REAL*4    PROCEDURE
Z*      000042  REAL*4    VARIABLE
INT     000000  INTEGER*2 PROCEDURE
HOL     000000  INTEGER*2 PROCEDURE
J       000046  INTEGER*2 VARIABLE
K       000050  INTEGER*2 VARIABLE
IOR     000000  INTEGER*2 PROCEDURE

COMMON BLOCK /GBUF/     LENGTH 020010

GBUF    000000  LOGICAL*1 ARRAY (B192)
C*      020000  REAL*4    VARIABLE
C*      020004  REAL*4    VARIABLE

COMMON BLOCK /LOULFT/   LENGTH 000020

/LL     000000  REAL*4    VARIABLE
/LL     000004  REAL*4    VARIABLE
/SHFT   000010  REAL*4    VARIABLE
/SHFT   000014  REAL*4    VARIABLE

COMMON BLOCK /DEBUG/    LENGTH 000002
```

```
IFLG    000000  INTEGER*2 VARIABLE

COMMON BLOCK  LAST/     LENGTH 000004

XLAST   000000  INTEGER*2 VARIABLE
YLAST   000002  INTEGER*2 VARIABLE
```

```
FORTRAN IV      V01C-03         FRI 28-SEP-79 09:26:29          PAGE 001
COPE=08K, UIC=[1,123]                                  ,TT0:=DK3:COMGRF.FTN

      C
      C
      C
      C
      C
0001          SUBROUTINE ADJUST(A,N)
      C
      C
      C       ALTERS THE PARAMETERS OF ALL CALLS TO 'LVECT','RDOT', SO AS
      C       TO REFLECT THE RESULTS OF SUBROUTINE 'SCAL'. THE ORIGINAL DEC
      C       J"-11 SUPPORT ROUTINE TOOK CARE OF THIS SCALING IN HARDWARE
      C       REGISTERS. THE COMTAL VERSION MUST PERFORM THIS FUNCTION IN
      C       SOFTWARE.
      C
      C
      C
0002          COMMON /SCALE/ SCL
      C
0003          DIMENSION A(6)
      C
0004          DO 10  I=1,N
0005          A(I) = A(I)*SCL
0006       10 CONTINUE
0007          RETURN
0008          END
```

```
FORTRAN IV      STORAGE MAP

NAME    OFFSET  ATTRIBUTES

A       000014  REAL*4    PARAMETER ARRAY (6)
N       000016  INTEGER*2 PARAMETER VARIABLE
I       000020  INTEGER*2 VARIABLE
```

```
COMMON BLOCK  SCALE     LENGTH 000004
SCL      000000  REAL*4    VARIABLE
FOR
```

```
FORTRAN IV      V01C-03          FRI 28-SEP-79 18:02:40          PAGE 001
CORE=08K, UIC=[1,123]                                DK3:VECTOR,TT0:=DK3:VECTOR

0001          SUBROUTINE UNITV (R,S,V)
0002          REAL R(3),S(3)
          C
          C
          C          COMPUTE UNIT VECTOR AND MAGNITUDE
          C
0003          V=SQRT(R(1)**2 + R(2)**2 + R(3)**2)
0004          IF(V.EQ.0.) RETURN
0005          DO 40 I=1,3
0007       40 S(I)=R(I)/V
          C
0008          RETURN
0009          END
```



```
FORTRAN IV      STORAGE MAP

NAME    OFFSET  ATTRIBUTES

R       000014  REAL*4    PARAMETER ARRAY (3)
S       000016  REAL*4    PARAMETER ARRAY (3)
V       000020  REAL*4    PARAMETER VARIABLE
SQRT    000000  REAL*4    PROCEDURE
I       000022  INTEGER*2 VARIABLE
```

```
FORTRAN IV      V01C-03          FRI 28-SEP-79 18:02:44          PAGE 001
CORE=08K, UIC=[1,123]                                DK3:VECTOR,TT0:=DK3:VECTOR

0001          SUBROUTINE DOTP (R,S,V)
0002          REAL R(3),S(3)
          C
          C
          C          COMPUTE DOT PRODUCT OF 2 VECTORS
          C
0003          V=0.
0004          DO 40 I=1,3
0005       40 V = V + R(I)*S(I)
          C
0006          RETURN
0007          END
```

```
FORTRAN IV      STORAGE MAP

NAME    OFFSET  ATTRIBUTES

R       000014  REAL*4    PARAMETER ARRAY (3)
S       000016  REAL*4    PARAMETER ARRAY (3)
J       000020  REAL*4    PARAMETER VARIABLE
I       000022  INTEGER*2 VARIABLE
```

```
FORTRAN IV      V01C-03         FRI 28-SEP-79 18:03:07          PAGE 001
CORE-08K, UIC=[1,123]                           DK3:VECTOR,TT0:=DK3:VECTOR

0001          SUBROUTINE CROSSP (R,S,T)
0002          REAL R(3),S(3),T(3)
              C
              C
              C         CROSS PRODUCT OF 2 VECTORS
              C
0003          T(1) = R(2)*S(3) - R(3)*S(2)
0004          T(2) = R(3)*S(1) - R(1)*S(3)
0005          T(3) = R(1)*S(2) - R(2)*S(1)
              C
0006          RETURN
0007          END
```

```
FORTRAN IV      STORAGE MAP

NAME    OFFSET  ATTRIBUTES

R       000014  REAL*4    PARAMETER ARRAY (3)
S       000016  REAL*4    PARAMETER ARRAY (3)
T       000020  REAL*4    PARAMETER ARRAY (3)
```

```
FORTRAN IV      V01C-03         WED 24-OCT-79 12:08:19          PAGE 001
COPE-08K, UIC=[1,123]                                  ,TT0:=DK3:ROTATE

      C     THIS ROUTINE DELIVERS THE SLANT RANGE AND CROSS RANGE VECTORS
      C     CALCULATED FROM THE ANGLES READ FROM THE INPUT DEVICE NUMBER 5
      C
      C     THE ANGLES ARE INPUT IN THE ORDER YAW,PITCH,ROLL IN DEGREES.
      C
      C     THE CALCULATIONS ARE BASED ON THE EQUATION FOUND IN THE SRC DOCUMENT
      C     SCR TR 77-284, DECEMBER 1977    RADAR IMAGE SIMULATION SYSTEM
      C                                     PDP 11/40 USER'S GUIDE
      C     F30602-77-C-0088
      C     APPENDIX C, PAGE C-1
      C
      C
      C
      C
0001        SUBROUTINE ROTATE (RS,RC)
0002        REAL         RS(3),RC(3)
0003        CONV = .0174533
0004        WRITE(5,10)
0005        READ(5,11)YAW,PITCH,ROLL
0006  10    FORMAT('INPUT YAW,PITCH,ROLL (REAL,SIGNED DEGREES):')
0007  11    FORMAT(F6.5,F6.5,F6.5)
0008        YAW=YAW*CONV
0009        PITCH=PITCH*CONV
0010        ROLL=ROLL*CONV
0011        RS(1)=COS(YAW)*COS(ROLL)-SIN(YAW)*SIN(PITCH)*SIN(ROLL)
0012        RS(2)=-SIN(YAW)*COS(PITCH)
0013        RS(3)=SIN(ROLL)*COS(YAW)+COS(ROLL)*SIN(YAW)*SIN(PITCH)
0014        RC(1)=COS(ROLL)*SIN(YAW)+COS(YAW)*SIN(ROLL)*SIN(PITCH)
0015        RC(2)=COS(YAW)*COS(PITCH)
0016        RC(3)=SIN(ROLL)*SIN(YAW)-COS(ROLL)*SIN(PITCH)*COS(YAW)
0017        RETURN
0018        END


FORTRAN IV      STORAGE MAP

NAME    OFFSET  ATTRIBUTES

RS      000014  REAL*4    PARAMETER ARRAY (3)
RC      000016  REAL*4    PARAMETER ARRAY (3)
```

CONV	000116	REAL*4	VARIABLE
YAW	000122	REAL*4	VARIABLE
PITCH	000126	REAL*4	VARIABLE
ROLL	000132	REAL*4	VARIABLE
COS	000000	REAL*4	PROCEDURE
SIN	000000	REAL*4	PROCEDURE

```
RT-11 FORTRAN IV        V01B-08A   MON 29-OCT-79 13:40:44       PAGE 001

0001          REAL D(256)
0002          INTEGER TTYI,TTYO,PARAFL(5),IMFILE(5),PARAM(5),IMAGE(5)
0003          DIMENSION RS(3),RC(3)
0004          DEFINE FILE 30(512,256,U,IAV)
0005          DEFINE FILE 31(1,256,U,JAV)
0006          WRITE(7,31)
0007          CALL ASSIGN(12,PARAFL,-1,'RDO')
0008          WRITE(7,32)
0009          CALL ASSIGN(31,PARAM,-1)
0010  30      FORMAT(/,' IMAGE FILE NAME ',/)
0011  31      FORMAT(/,' PARAMETER FILE NAME ',/)
0012  32      FORMAT(/,' ENTER DESTINATION FILE NAME',/)
0013          READ(12)NURDS,NRECS,RMAX,SRI,CRI,(RS(I),I=1,3),
             X     (RC(I),I=1,3)
0014          REWIND 12
0015          WRITE(7,50)NURDS,NRECS,RMAX,SRI,CRI,
             X     (RS(I),I=1,3),(RC(I),I=1,3)
0016  50      FORMAT(' NURDS = ',I5,5X,'NRECS = ',I5,' RMAX = ',F12.3,5X,
             X     'SRI = ',F8.3,5X,'CRI = ',F8.3,/' RS(I):   ',3(F8.3,5X),
             X     'RC(I):   ',3(F8.3,5X))
0017          WRITE(31'1)NURDS,NRECS,RMAX,SRI,CRI,(RS(I),I=1,3),
             X     (RC(I),I=1,3)
0018          WRITE(7,30)
0019          CALL ASSIGN(11,IMFILE,-1,'RDO')
0020          WRITE(7,32)
0021          CALL ASSIGN(30,IMAGE,-1)
0022          DO 45 I9=1,NRECS
0023          READ(11) (D(K),K=1,NURDS)
0024          WRITE(30'I9) (D(K),K=1,NURDS)
0025  45      CONTINUE
0026          REWIND 11
0027          STOP
0028          END


RT-11 FORTRAN IV        STORAGE MAP

NAME    OFFSET  ATTRIBUTES

D       000006  REAL*4    ARRAY (256)
PARAFL  002006  INTEGER*2 ARRAY (5)
```

```
IMFILE   002020  INTEGER*2 ARRAY (5)
PARAM    002032  INTEGER*2 ARRAY (5)
IMAGE    002044  INTEGER*2 ARRAY (5)
RS       002056  REAL*4    ARRAY (3)
RC       002072  REAL*4    ARRAY (3)
TTYI     002436  INTEGER*2 VARIABLE
TTYO     002440  INTEGER*2 VARIABLE
IAU      002442  INTEGER*2 VARIABLE
JAU      002444  INTEGER*2 VARIABLE
ASSIGN   000000  REAL*4    PROCEDURE
NWRDS    002446  INTEGER*2 VARIABLE
NRECS    002450  INTEGER*2 VARIABLE
RMAX     002452  REAL*4    VARIABLE
SRI      002456  REAL*4    VARIABLE
CRI      002462  REAL*4    VARIABLE
I        002466  INTEGER*2 VARIABLE
I9       002470  INTEGER*2 VARIABLE
K        002472  INTEGER*2 VARIABLE
$
```

MISSION
of
Rome Air Development Center

RADC plans and executes research, development, test and selected acquisition programs in support of Command, Control, Communications and Intelligence (C3I) activities. Technical and engineering support within areas of technical competence is provided to ESD Program Offices (POs) and other ESD elements. The principal technical mission areas are communications, electromagnetic guidance and control, surveillance of ground and aerospace objects, intelligence data collection and handling, information system technology, ionospheric propagation, solid state sciences, microwave physics and electronic reliability, maintainability and compatibility.