IRIS R7.5

RELEASE NOTES

COPY (Not Up to Date)

POINT 4

DATA CORPORATION

POINT 4 Data Corporation
2569 McCabe Way, Irvine, CA 92714

June 30, 1982

IRIS 7.5 RELEASE NOTES

CONTENTS

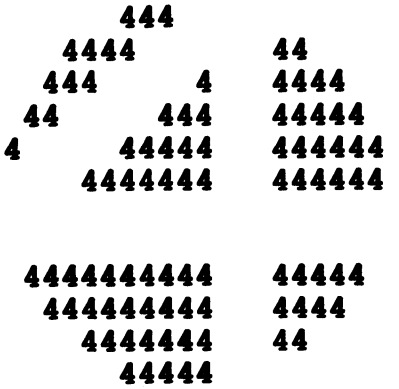
The following materials may be included as attachments:

a. IRIS on paper tape

b. One or more Front-end packages (listing and objective papertape of SOV, BZUP, and DDCOPY) specific to your LU/0 disc controller and drive

c. Peripherals Handbook

POINT 4 Data Corporation



TECHNICAL MEMORANDUM

TO: ALL IRIS USERS

FROM: SYSTEMS SUPPORT

DATE: January 29, 1982

SUBJ: DIFFERENCES BETWEEN R7.4 AND R7.5 ON THE MARK 5

The following is a list of the differences between R7.4 and R7.5 on the MARK 5.

1. The following enhancements or corrections have been made to the system:

   BASIC - Improvement in the compilation of the "IF" operator

   CLEANUP - First active file in a sequence now will be forced to a cylinder boundary
   - Allows expansion of a logical unit
   - Prevents TRAP 5 if bad block occurs in last active file

   COPY - Prevents TRAP 5 when copying Formatted or BASIC files and will not TRAP when copying extended textfiles

   DISCSUBS - Prevents TRAP 0 at 6010 in various processors
   - Corrects improper allocation of processors
   - Prevents crash when reading tapes at 1600 BPI

   DSP - Immediately updates; calls FLUSH

   EDIT - Prevents TRAP 0 in alphabetical LIBR after EDIT of large file

   $LPTP - Prevents hanging the system when an interrupt is generated by the printer before the first OPEN is issued to the printer

   PORT - Lists user's port under MONITOR option, and flags user's port with an asterisk (*) when ALL option is used

   REX - Corrects handling of maximum (7) nested subroutine calls

RUN - Prevents TRAP 0 at 0 in the case when too many characters are printed without a CR
- Prevents TRAP or system hanging on ESC with printer open
- Handles terminals that generate greater than 6 characters through $TERMS (e.g., VT100)

RUNMAT - Prevents killing another user's active file; also prevents problems with MAT ZERO

SIR - Calculates the number of active file blocks correctly
- Allows loading of peripheral drivers greater than 2 blocks long

VERIFY - Puts terminating null on output

2. New CONFIG and BLOCKCOPY (see Peripherals Handbook for latest information)

3. New BYE processor with upgraded support for auto program start

4. Upgraded support for STYLUS

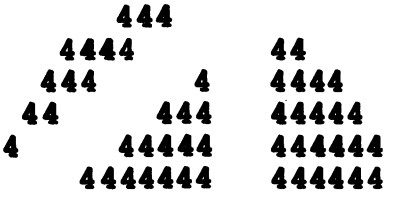
5. New MIGHTY MUX handler ($MMUX) to support TYPIST, $COM and to correct known problems

6. IRIS support for TYPIST

7. Software to support the LCM (LOTUS Cache Memory)

8. Software to support serial interprocessor communication applications ($COM).

POINT 4 Data Corporation



T E C H N I C A L
M E M O R A N D U M

TO: ALL IRIS USERS

FROM: SYSTEMS SUPPORT

DATE: February 25, 1982

SUBJ: DIFFERENCES BETWEEN R7.4 AND R7.5 ON THE MARK 3

The following is a list of the differences between R7.4 and R7.5 on the MARK 3.

1. The following enhancements or corrections have been made to the system:

   BASIC - Improvement in the compilation of the "IF" operator

   CLEANUP - First active file in a sequence now will be forced to a cylinder boundary
   - Allows expansion of a logical unit
   - Prevents TRAP 5 if bad block occurs in last active file

   COPY - Prevents TRAP 5 when copying Formatted or BASIC files and will not TRAP when copying extended textfiles

   DISCSUBS - Prevents TRAP 0 at 6010 in various processors
   - Corrects improper allocation of processors

   DSP - Immediately updates; calls FLUSH

   EDIT - Prevents TRAP 0 in alphabetical LIBR after EDIT of large file

   PORT - Lists user's port under MONITOR option, and flags user's port with an asterisk (*) when ALL option is used

   REX - Corrects handling of maximum (7) nested subroutine calls
   - Uses TOPW to determine size of memory

RUN - Prevents TRAP 0 at 0 in the case when too many characters are printed without a CR
- Prevents TRAP or system hanging on ESC with printer open
- Handles terminals that generate greater than 6 characters through $TERMS (e.g., VT100)

RUNMAT - Prevents killing another user's active file; also prevents problems with MAT ZERO

SIR - Calculates the number of active file blocks correctly
- Allows loading of peripheral drivers greater than 2 blocks long

VERIFY - Puts terminating null on output

2. New BYE processor with upgraded support for auto program start

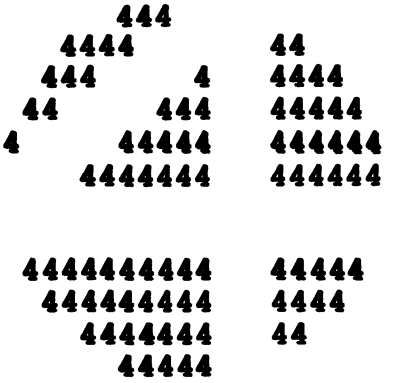
3. Upgraded support for STYLUS

4. New MIGHTY MUX handler ($MMUX) to support TYPIST, $COM and to correct known problems

5. IRIS support for TYPIST

6. Software to support serial interprocessor communication applications ($COM).

POINT 4 Data Corporation



TECHNICAL MEMORANDUM

TO: ALL IRIS USERS

FROM: SYSTEMS SUPPORT

DATE: April 14, 1982

SUBJ: 7.5 UPDATE PROCEDURE FOR MARK 5

NOTE: FOR MARK 5 SYSTEMS ONLY. (FOR MARK 3 SYSTEMS SEE UPDATE PROCEDURE FOR MARK 3)

I. PREPARATION FOR 7.5 UPGRADE

1. Update your IRIS 7.5 Peripherals Handbook with the new cover sheet and the new Disc Specification sheets.

2. Back-up your system and data.

II. EXTRACT CURRENT SYSTEM SETUP INFORMATION

1. IPL your present (7.4) system

2. CONFIG
   a. Enter DSP and FCONFIG
   b. Write down the contents of the following locations:
      i. 400 thru 405
      ii. 601 thru 615
      iii. 617 thru 620
      iv. 1000 thru the first occurrence of 77377
      v. 1400 thru the first occurrence of 177777 followed by 77377
      vi. 16000 thru 16377 (if you get the message NO SUCH ADDRESS just continue)
      vii. 16400 thru 16777 (if you get the message NO SUCH ADDRESS just continue)

3. ACCOUNTS
   a. Enter DSP and F0/ACCOUNTS
   b. H (to get the header)
   c. Change word 10 to 31
   d. X (to exit DSP)
   e. Copy the accounts file to a logical unit that is either on a different drive or on a different platter than logical unit zero. Use the filename LUZEROACC, e.g.,
      COPY X/LUZEROACC=0/ACCOUNTS

4. MUX
   a. If you are using a POINT 4 MIGHTY MUX go to step II4d
   b. If you are not using a POINT 4 MIGHTY MUX change the filename of your MUX driver to the same filename without the $ sign; then copy it to the same logical unit to which you copied LUZEROACC.
   c. Go to step 5
   d. If you are using a POINT 4 MIGHTY MUX enter DSP and F$MMUX
   e. D10201
   f. D (contents -1 of location 10201) i.e., if cell 10201 contains 11234, then the command would be D11233
   g. Write down the contents from the location found in step II4f thru the second occurrence of 177777

5. USERID
   a. Copy USERID to the same logical unit to which LUZEROACC was copied. Use filename USERID.OLD, e.g.,
      COPY X/USERID.OLD=0/USERID

6. For each lineprinter driver that is enabled (prefixed with a $)
   a. Enter DSP and F (the filename)
   b. H (to get header)
   c. Change word 10 from 36 to 3
   d. X (to exit DSP)
   e. Change the filename to the same filename without the $ using the CHANGE processor
   f. Copy the lineprinter driver to the same logical unit to which LUZEROACC was copied. Use the same filename without the $ and suffix with .OLD.

7. RUN call table
   a. Enter DSP and FRUN
   b. D540
   c. Write down the contents of 540 thru the first occurrence of 177777

8. $ sign files
   a. Run a LIBR of the $ files from the manager's account
      LIBR @$ (then write them down)
        or
      LIBR @$[$LPT] (to output to the lineprinter)

9. NUMBER OF CYLINDERS of PRESENT LOGICAL UNIT ZERO
   a. Enter DSP and FCONFIG
   b. D1404
   c. FREX
   d. D (contents of 1404 in CONFIG) (see step II9b.) The first word is the number of cylinders (given in octal) of your present logical unit.

10. LCM.PARAMS
   a. Copy LCM.PARAMS to the same logical unit to which LUZEROACC was copied. Use the filename LCM.PARAMSZERO, e.g.,
      COPY X/LCM.PARAMSZERO=0/LCM.PARAMS

III. OVERLAY 7.5

1. PUT on "IRIS" R7.5 via CTU or a disc-to-disc copy program. NOTE: If you received R7.5 on disc, copy only logical unit zero, not the whole disc unless your present logical unit zero is the whole disc.

IV. ADD SAVED SYSTEM INFORMATION TO 7.5 SYSTEM

1. IPL into a minimum config

2. CONFIG
   a. Enter DSP and FCONFIG
   b. Enter the old contents into their respective locations from step II2b. WARNING: do not change locations 600 and 616.
   c. See APPENDIX A

3. MUX
   a. If you are using a POINT 4 MIGHTY MUX go to step IV3d
   b. If you are not using a POINT 4 MIGHTY MUX, copy your MUX back from the logical unit you copied it to in step II4b, then change it back to a $ sign file
   c. Go to step 4
   d. If you are using a POINT 4 MIGHTY MUX, enter DSP and F$MMUX
   e. Dump contents -1 of location 10201, i.e., if cell 10201 contains 11234, then the command would be D11233.
      NOTE: This will not be the same address you got in step II4f
   f. Enter the old contents from step II4g starting at the location found in step IV3e

4. $ sign files
   a. Use the CHANGE processor to change each file listed in step II8 to a $ sign file EXCEPT for lineprinter drivers

5. SHUTDOWN and perform a full config IPL

6. Install the partition to which you copied LUZEROACC

7. Run the BASIC program ACCRESTORE

8. For each lineprinter driver copied to the same logical unit that you copied LUZEROACC:
   a. Copy back to logical unit zero using the same filename as on the other logical unit
   b. Enter DSP and F (the filename)
   c. H (to get header)
   d. Change word 10 from 77003 to 36
   e. X (to exit DSP)
   f. Change the filename to its original filename (with the $ prefix and without the .OLD suffix) using the CHANGE processor

9. Patch LPTP73-07
   a. Enter this patch into all lineprinter drivers that you have enabled AND which have their own I/O board. WARNING: This does NOT go into lineprinter drivers that run thru the $MMUX

10. RUN call table
   a. Enter DSP and FRUN
   b. D540
   c. Check that all the words you wrote down in step II7c are still there, enter any words that are not there. NOTE: There will probably be some new words in the table
   d. Be sure that there is a 177777 to terminate the table

11. ACTIVE FILE SIZE
   a. Enter DSP and FCONFIG
   b. D401 (PSIZ)
   c. If equal to 14400 (octal) goto step IV,12
   d. Divide PSIZ by 400 (remember: octal)
   e. FREX
   f. Enter the quotient at location 22426
   g. FPHA (or $PHA)
   h. Enter the quotient at location 10226

12. LCM.PARAMS
   a. Copy LCM.PARAMSZERO back to logical unit zero. Use the filename LCM.PARAMS, e.g.,
      COPY 0/LCM.PARAMS!=X/LCM.PARAMSZERO
   b. If your system does not have an LCM go to step 13
   c. If your system does have an LCM, run the BASIC program LCMCONFIGURE

13. SHUTDOWN and perform a new IPL

14. Now your IRIS system has been upgraded to the supported Revision Level 7.5

APPENDIX A

Refer to the Disc Specification Sheets for your system and enter any set-up parameters given for your primary and any secondary controllers.

Some of the BZUPs and SOVs in CONFIG have been moved. If you happen to have one of the configurations listed below you will have to change the LUFIX and BZUP addresses in the disc table in config starting at address 1400.

```
ENTRY #29
            DISC ID          S12CMD
            CONTROLLER       MCT SMC12
            DRIVE            CDC 9448
                  LUFIX ADDRESS
                           OLD = 35424
                           NEW = 35324
                  BZUP ADDRESS
                           OLD = 35303
                           NEW = 35203

ENTRY #34
            DISC ID          MC9F50
            CONTROLLER       MCT SMC-902
            DRIVE            FUJITSU 50 MB
                  LUFIX ADDRESS
                           OLD = 7024
                           NEW = 7024 (same)
                  BZUP ADDRESS
                           OLD = 15523
                           NEW = 42303

ENTRY #35
            DISC ID          QUECMD
            CONTROLLER       QUENTIN N6010
            DRIVE            CDC 9448
                  LUFIX ADDRESS
                           OLD = 40074
                           NEW = 36224
                  BZUP ADDRESS
                           OLD = 13614
                           NEW = 13614 (same)

ENTRY #37
            DISC ID          EDSO80
            CONTROLLER       LOTUS 700
            DRIVE            OKIDATA 80 MB
                  LUFIX ADDRESS
                           OLD = 37154
                           NEW = 37224
                  BZUP ADDRESS
                           OLD = 10314
                           NEW = 10314 (same)

ENTRY #38
            DISC ID          EDSC80
            CONTROLLER       LOTUS 700
            DRIVE            BALL BD 80 or CALCOMP T-82
                             or CDC 80 MB (9762)
                  LUFIX ADDRESS
                           OLD = 37154
                           NEW = 42424
                  BZUP ADDRESS
                           OLD = 10314
                           NEW = 41664
```

POINT 4
DATA CORPORATION

SOFTWARE CHANGE ORDER

software production manager ONLY

STATUS _____

PROD. SYSTEM
UPDATE BY _____ DATE _____

SCO # _____ DATE _____

MASTER FILE
UPDATE BY _____ DATE _____

MASTER
FILENAME _____

PATCH # LPTP73-07

PRODUCT $LPTP

DETAIL _____

ASM DATE 1-25-79 RELEASE # 7.3

UPDATE DATE 6-29-79 BY SCM

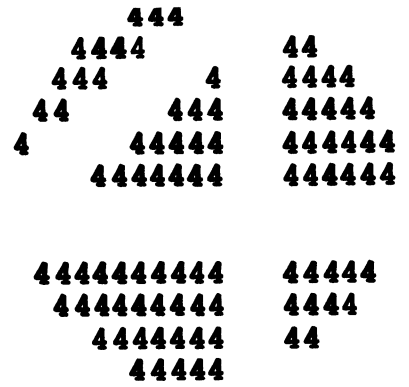
REVIEW DATE _____ BY _____

PR # _____ PROBLEM Any interrupts from the printer before the first open can hang the system (E.G. just powering it on).

SPECIAL INSTRUCTIONS (other products?) This completely replaces LPTP73-02. The line below marked * make sure LPTP73-02 is removed properly.

LOCATION (OCTAL)	NEW CONTENTS (OCTAL AND/OR SYMBOLIC)	COMMENTS (DESCRIBE SOLUTION)	OLD CONTENTS
10224	425		25005
10225	21004 *		21004
10227	1400 *		1400
10246	401		404
10251	151015		0
2	2771		20777
3	25005		61117
10254	751		737
10351	642		642

POINT 4 Data Corporation



T E C H N I C A L
M E M O R A N D U M

TO: ALL IRIS USERS

FROM: SYSTEMS SUPPORT

DATE: February 26, 1982

SUBJ: 7.5 UPDATE PROCEDURE FOR MARK 3

NOTE: FOR MARK 3 SYSTEMS ONLY. (FOR MARK 5 SYSTEMS SEE UPDATE PROCEDURE FOR MARK 5)

I. PREPARATION FOR 7.5 UPGRADE

1. Back-up your system and data

II. EXTRACT CURRENT SYSTEM SETUP INFORMATION

1. IPL your present (7.4) system

2. CONFIG
   a. Enter DSP and FCONFIG
   b. Write down the contents of the following locations:
      i. 400 thru 405
      ii. 601 thru 615
      iii. 617 thru 620
      iv. 1000 thru the first occurrence of 77377
      v. 1400 thru the first occurrence of 177777 followed by 77377
      vi. 16000 thru 16377 (if you get the message NO SUCH ADDRESS just continue)
      vii. 16400 thru 16777 (if you get the message NO SUCH ADDRESS just continue)

3. ACCOUNTS
   a. Enter DSP and F0/ACCOUNTS
   b. H (to get the header)
   c. Change word 10 to 31
   d. X (to exit DSP)
   e. Copy the accounts file to a logical unit that is either on a different drive or on a different platter than logical unit zero. Use the filename LUZEROACC, e.g.,

      COPY X/LUZEROACC=0/ACCOUNTS

4. $MMUX
   a. Enter DSP and F$MMUX
   b. D10201
   c. D (contents -1 of location 10201) i.e., if cell 10201 contains 11234 then the command would be D11233
   d. Write down the contents from the location found in step II4c thru the second occurrence of 177777

5. USERID
   a. Copy USERID to the same logical unit to which LUZEROACC was copied. Use filename USERID.OLD, e.g.,

```
COPY X/USERID.OLD=0/USERID
```

6. For each lineprinter driver that is enabled (prefixed with a $)
   a. Enter DSP and F (the filename)
   b. H (to get header)
   c. Change word 10 from 36 to 3
   d. X (to exit DSP)
   e. Change the filename to the same filename without the $ using the CHANGE processor
   f. Copy the lineprinter driver to the same logical unit to which LUZEROACC was copied. Use the same filename without the $ and suffix with .OLD

7. RUN call table
   a. Enter DSP and FRUN
   b. D540
   c. Write down the contents of 540 thru the first occurrence of 177777

8. $ sign files
   a. Run a LIBR of the $ files from the manager's account

```
LIBR @$     (then write them down)
  or
LIBR @$[$LPT]  (to output to the lineprinter)
```

9. NUMBER OF CYLINDERS of PRESENT LOGICAL UNIT ZERO
   a. Enter DSP and FCONFIG
   b. D1404
   c. FREX
   d. D (contents of 1404 in CONFIG) (see step II9b.) The first word is the number of cylinders (given in octal) of your present logical unit zero.

III. OVERLAY 7.5

1. PUT on "IRIS" R7.5 via CTU or a disc-to-disc copy program. NOTE: If you received R7.5 on disc, copy only logical unit zero, not the whole disc unless your present logical unit zero is the whole disc.

IV. ADD SAVED SYSTEM INFORMATION TO 7.5 SYSTEM

1. IPL into a minimum config

2. CONFIG
   a. Enter DSP and FCONFIG
   b. Enter the old contents into their respective locations from step II2b. WARNING: do not change locations 600 and 616.

3. MUX
   a. Enter DSP and F$MMUX
   b. Dump contents -1 of location 10201 (see step II4c). NOTE: This will not be the same address you got in step II4c.
   c. Enter the old contents from step II4d starting at the location found in step IV3b.

4. $ sign files
   a. Use the CHANGE processor to change each file listed in step II8 to a $ sign file EXCEPT for lineprinter drivers

5. SHUTDOWN and do a full config IPL

6. Install the partition to which you copied LUZEROACC

7. Run the BASIC program ACCRESTORE

8. For each lineprinter driver copied to the same logical unit that you copied LUZEROACC:
   a. Copy back to logical unit zero using the same filename as on the other logical unit
   b. Enter DSP and F (the filename)
   c. H (to get header)
   d. Change word 10 from 77003 to 36
   e. X (to exit DSP)
   f. Change the filename to its original filename (with the $ prefix and without the .OLD suffix) using the CHANGE processor

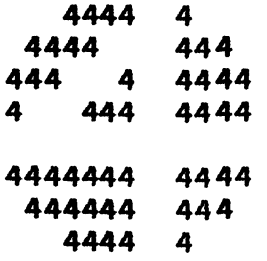
9. RUN call table
   a. Enter DSP and FRUN
   b. D540
   c. Check that all the words you wrote down in step II7c are still there, enter any words that are not there. NOTE: There will probably be some new words in the table
   d. Be sure that there is a 177777 to terminate the table

10. ACTIVE FILE SIZE
    a. Enter DSP and FCONFIG
    b. D401 (PSIZ)
    c. If equal to 14400 (octal) goto step IV,11
    d. Divide PSIZ by 400 (remember: octal)
    e. FREX
    f. Enter the quotient at location 22366
    g. FPHA (or $PHA)
    h. Enter the quotient at location 10226

11. SHUTDOWN and perform a new IPL

12. Now your IRIS system has been upgraded to the supported Revision Level 7.5

POINT 4 Data Corporation



T E C H N I C A L M E M O R A N D U M

TO: IRIS 7.5 Users (Mark 5, Mark 8)

FROM: Customer Support

DATE: December 29, 1981

SUBJ: IRIS Utilization of Mark 8 features ($MK8)

A driver exists that allows IRIS to utilize the macro intruction features of the Mark 8 computer. This driver is called $MK8. $MK8 takes the place of $DEC and must not be enabled at the same time as $DEC is enabled.

If you are running IRIS on a Mark 8 computer, change the file "MK8" to "$MK8" and change "$DEC" to "DEC". Then IPL to activate the new driver.

** PRODUCT LIST **

NOV 20, 1981 15:15

FILE NO.	NAME	RELEASE	ASM DATE	TYPE	REV DATE	GROUP
1	REX	7.5	06-02-78	OBJECT	11-16-81	1
2	SYSGEN	7.5	06-02-78	OBJECT	11-16-81	1
3	DBUG.CTU	7.5	06-24-80	OBJECT	11-16-81	1
4	DISCSUBS (#1)	7.5	06-01-77	OBJECT	05-07-80	1
5	SCOPE	7.5	02-07-80	OBJECT	02-07-80	1
6	BYE	7.5	11-06-81	OBJECT	11-16-81	1
7	DSP	7.5	04-25-78	OBJECT	07-18-79	1
8	CONFIG	7.5	11-11-81	OBJECT	11-16-81	1
9	BLOCKCOPY	7.5	04-28-81	OBJECT	11-16-81	1
10	CHANGE	7.5	01-26-78	OBJECT	01-26-80	1
11	CLEANUP	7.5	12-12-78	OBJECT	01-26-80	1
12	CLEANUPX	7.5	12-12-79	OBJECT	11-16-81	1
13	CONVERT	7.5	06-10-77	OBJECT	01-26-80	1
14	COPY	7.5	06-10-77	OBJECT	07-18-79	1
15	INSTALL	7.5	02-09-78	OBJECT	07-18-79	1
16	KILL	7.5	05-02-77	OBJECT	05-02-77	1
17	MAIL	7.5	05-02-77	OBJECT	05-02-77	1
18	MESSAGES	7.5	06-01-77	OBJECT	02-16-78	1
19	PORT	7.5	05-10-78	OBJECT	07-23-78	1
20	PROTECT	7.5	04-16-80	OBJECT	04-16-80	1
22	QUERY	7.5	05-02-77	OBJECT	07-28-78	1
26	REHASH	7.5	05-31-78	OBJECT	05-31-78	1
27	REMOVE	7.5	05-02-77	OBJECT	05-31-78	1
28	SAVE	7.5	04-17-80	OBJECT	04-22-80	1
29	SHUTDOWN	7.5	04-19-77	OBJECT	07-18-79	1
30	VERIFY	7.5	04-17-80	OBJECT	04-17-80	1
31	LIBR	7.5	06-01-77	OBJECT	01-26-80	1
102	LCMACTIVATE	7.5	10-30-81	TEXT	11-16-81	1
103	LCMMODIFY	7.5	10-30-81	TEXT	11-16-81	1

FILE NO.	NAME	RELEASE	ASM DATE	TYPE	REV DATE	GROUP
32	FORMAT	7.5	05-02-77	OBJECT	05-02-77	2
33	RUNMAT	7.5	04-14-80	OBJECT	04-14-80	2
34	BASIC	7.5	04-15-80	OBJECT	05-31-80	2
35	RUN	7.5	04-11-80	OBJECT	04-11-80	2
36	DISCSUBS (#2)	7.5	07-25-78	OBJECT	05-07-80	2
37	ACCOUNTLIST	7.5	07-21-76	TEXT	07-21-76	2
38	ACCRESTORE	7.5	11-06-81	TEXT	11-16-81	2
39	ALOAD	7.5	11-14-80	TEXT	11-16-81	2
40	BASICTEST	7.5	01-05-77	TEXT	01-05-77	2
41	BUILDXF	7.5	07-10-76	TEXT	05-25-77	2
42	EXERCISER	7.5	06-14-79	TEXT	06-14-79	2
43	EXTRAPORT	7.5	03-31-78	TEXT	09-22-78	2
44	GUIDE	7.5	01-24-79	TEXT	07-18-79	2
45	GUIDE.LPT	7.5	01-24-79	TEXT	07-18-79	2
46	GUIDE.LU	7.5	01-24-79	TEXT	07-18-79	2
47	GUIDE.BLOCKCPY	7.5	01-24-79	TEXT	07-18-79	2
48	LCMCHECK	7.5	10-30-81	TEXT	11-16-81	2
49	LCMCONFIGURE	7.5	10-30-81	TEXT	11-16-81	2
50	LPTDIAG	7.5	03-21-80	TEXT	05-07-80	2
51	LPTDIAG 2	7.5	03-21-80	TEXT	03-21-80	2
52	LPTDIAG 3	7.5	03-21-80	TEXT	03-21-80	2
53	LPTDIAG 4	7.5	03-21-80	TEXT	03-21-80	2
54	MTUTL	7.5	--------	TEXT	--------	2
55	RETRY	7.5	05-10-77	TEXT	05-10-77	2
56	SETTIME	7.5	--------	TEXT	--------	2
57	TAPETESTSW	7.5	10-30-81	TEXT	11-16-81	2
58	UTILITY	7.5	07-30-73	TEXT	01-31-77	2

FILE NO.	NAME	RELEASE	ASM DATE	TYPE	REV DATE	GROUP
59	ASSEMBLE	7.5	06-10-77	OBJECT	01-26-80	4
60	EDIT	7.5	06-08-77	OBJECT	01-26-80	4
61	DISCSUBS (#3)	7.5	06-13-77	OBJECT	01-26-80	4
62	DEFINITIONS	7.5	10-30-81	TEXT	11-16-81	4
63	PZ	7.5	10-30-81	TEXT	11-16-81	4
64	RMDTC	7.5	10-21-81	TEXT	11-16-81	4
65	SYMBOLS	7.5	05-02-77	TEXT	05-02-77	4
66	L.LCMDEFS	7.5	10-30-81	TEXT	11-16-81	4
67	LCM.PARAMS	7.5	10-30-81	TEXT	11-16-81	4

FILE NO.	NAME	RELEASE	ASM DATE	TYPE	REV DATE	GROUP
68	$RTC	7.5	05-03-77	OBJECT	05-03-77	12
69	$MMUX	7.5	11-05-81	OBJECT	11-16-81	12
70	$DGMX	7.5	01-07-78	OBJECT	01-07-78	12
71	$PHA	7.5	06-10-77	OBJECT	01-20-79	12
72	$TTY	7.5	01-20-79	OBJECT	01-20-79	12
73	$PTR	7.5	05-06-77	OBJECT	05-06-77	12
74	$PTP	7.5	05-05-77	OBJECT	05-05-77	12
75	$PTM	7.5	05-05-77	OBJECT	05-05-77	12
76	$DEC	7.5	01-26-78	OBJECT	01-26-78	12
77	$LPTM	7.5	01-22-79	OBJECT	01-26-80	12
78	$LPTP	7.5	01-25-79	OBJECT	07-18-79	12
79	$COMA	7.5	03-20-81	OBJECT	11-16-81	12
80	$COMD	7.5	03-21-81	OBJECT	11-16-81	12
81	LPTDIAGDRIVER	7.5	03-21-80	OBJECT	05-07-80	12
82	$LCM	7.5	10-30-81	OBJECT	11-16-81	12
83	$MTAO	7.5	02-28-79	OBJECT	02-28-79	12
84	$MTAS	7.5	02-28-79	OBJECT	02-28-79	12
85	TERMS	7.5	04-29-81	OBJECT	11-16-81	12
86	TERM (ADM1)	7.5	04-29-81	OBJECT	11-16-81	12
87	TERM (ADM2)	7.5	04-29-81	OBJECT	11-16-81	12
88	TERM (ADM3)	7.5	04-29-81	OBJECT	11-16-81	12
89	TERM (B100)	7.5	02-10-80	OBJECT	11-16-81	12
90	TERM (DGC)	7.5	04-28-81	OBJECT	11-16-81	12
91	TERM (DM1520)	7.5	04-29-81	OBJECT	11-16-81	12
92	TERM (DM1521)	7.5	04-29-81	OBJECT	11-16-81	12
93	TERM HASL 2000	7.5	05-21-79	OBJECT	05-21-79	12
94	TERM HASL 1500	7.5	01-24-80	OBJECT	01-24-80	12
95	TERM TV 912	7.5	04-29-81	OBJECT	11-16-81	12

97	TERMINET	7.5	12-26-77	OBJECT	12-26-77	12
98	TERM ACT 5	7.5	04-29-81	OBJECT	11-16-81	12
99	TERM ADDS 25	7.5	05-28-80	OBJECT	11-16-81	12
100	TERM VT 100	7.5	11-03-80	OBJECT	11-16-81	12
101	TERM VT 52	7.5	11-03-81	OBJECT	11-16-81	12

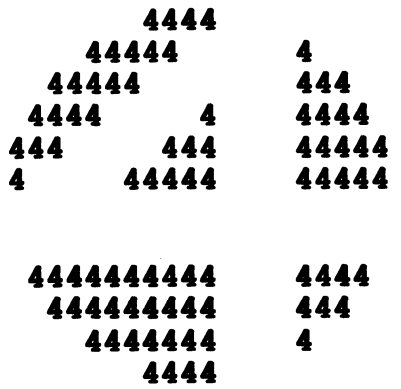
FILE NO.	NAME	RELEASE	ASM DATE	TYPE	REV DATE	GROUP
104	MUX(300)DP	7.5	11-08-75	OBJECT	11-08-75	16
105	MUX(310)DP	7.5	06-27-79	OBJECT	06-27-79	16

** PRODUCT LIST **

NOV 20, 1981 15:26

FILE NO.	NAME	RELEASE	ASM DATE	TYPE	REV DATE	GROUP
21	O. DMSTB	7.5	10-21-81	OBJECT	11-16-81	13
23	RADOC	7.5	10-21-81	OBJECT	11-16-81	13
24	RADSC	7.5	10-21-81	OBJECT	11-16-81	13
25	RCDTC	7.5	10-21-81	OBJECT	11-16-81	13

POINT 4 Data Corporation



T E C H N I C A L
M E M O R A N D U M

TO: ALL IRIS USERS

FROM: IRIS CUSTOMER SUPPORT

DATE: February 3, 1982

SUBJ: SOFTWARE RECEIVED ON DISC PACK

To any user who receives software (programs, data, IRIS, etc.) on a disc pack, we would like to recommend that the procedures in this memo be followed.

The problems, which we hope to prevent by this memo, can result in telephone calls to us such as:

> "I'd like to report a problem with the disc you sent me. It worked fine the first couple of times we used it, but today it has been giving me trap messages and often now, it won't even IPL".

The reason for this problem is that the pack was formatted and the data was written on another system and not on your own. Possible differences in drive tolerance, head alignment and temperature response make it impossible to guarantee that the data will be usable at all sites. In most cases, you will have no trouble with the pack sent. The procedure given here is to assure that problems do not develop later.

PROCEDURE:

1. Get a scratch pack which has been formatted in your system.

2. Feel the outside temperature of the disc we sent. If it is noticeably cooler or warmer than your own discs, let it sit with your discs until the temperature adjusts.

3. Bring DDCOPY into core (or any disc-to-disc copy program). If you have no other discs or papertapes to accomplish this, you may have to IPL our pack as a last resort. If so, use it only to SHUTDOWN to DDCOPY.

4. Copy our pack to your formatted scratch pack.

5. Put our pack away in a safe place as a back-up of the original software sent. Never use it except to make new copies from it.

I R I S 7 . 5 S Y S G E N L O G

Company Name _____ Date _____

Address _____ Name #1 _____

City, State _____ Name #2 _____

Telephone (_____)_____ IRIS Revision_____

Performed by _____ At _____

Disc controller: _____ Disc drive: _____

Refer to Section 2 of the IRIS Manager Reference Manual for detailed information on the System Generation (SysGen) procedure. Where there are differences, this log supersedes the manual.

=========> Run hardware diagnostics (initial when run):

_____ CPU Exerciser
_____ Power Fail Auto Restart Test
_____ Memory Address Test (all memory)
_____ Memory Checkerboard Test (all memory)
_____ Disc Reliability Test
_____ POINT 4 Multiplexer Test (including Q-Test)

NOTE: If using an MIGHTY MUX with T-option enabled, set the master terminalto baud rate 9600.

========> Use the POINT 4 binary loader (described in an Appendix of the IRIS Manager Reference Manual) to load the following into core:

Filename	Type	Asm Date	Pun Date	Comments
REX	-	_____	_____	(TEX for R4)_____
SYSGEN	-	_____	_____	_____
DBUG	-	_____	_____	_____
BZUP	-	_____	_____	(expect 73077 halt)_____
SOV	-	_____	_____	_____

=========> RESET and START at LSYSL if it doesn't start by itself. (LSYSL = 26000 for R4, 34000 for R7.2 & later releases).

DISCSUBS #1	-	_____	_____	_____
SCOPE	33400	_____	_____	_____
BYE	33400	_____	_____	_____
DSP	77400	_____	_____	_____

=========> Set all CPU switches down and press CTRL C.
The system will come up in a minimum configuration IPL with the message "NO CONFIG FILE".

Log on as "MANAGER" and use PLOAD to load the following tapes in the order listed:

CONFIG	77001			
$RTC	77001			(note #1)
$MMUX	77001			
$DGMX	77001			
$PHA	77001			
$TTY	77001			
$PTR	36			
$PTP	36			
$PTM	36			
$DEC	77001			(note #4)
$MK8	77001			(note #5)
LPTM	36			
LPTP	36			
$LPT	36			(note #2)
LPTDIAGDRIVER	36			
$COM.A	36			
$COM.D	36			
$MTA0	36			
$MTAS	77001			
$LCM	77001			
$TERMS.75	77001			
$TERMADM1.75	77001			
$TERMADM2.75	77001			
$TERMADM3.75	77001			
$TERMB100	77001			
$TERMDGC.75	77001			
$TERMDM1520.75	77001			
$TERMDM1521.75	77001			
$TERMH2000	77001			
$TERMH1500	77001			
$TERMTV912.75	77001			
$TERMTV950.75	77001			
$TERMINET	77001			
$TERMACT5.75	77001			
$TERMADDS25.75	77001			
$TERMVT100	77001			
$TERMVT52	77001			
BLOCKCOPY	77003			
CHANGE	33401			
CLEANUP	77401			
CLEANUPX	77401			
CONVERT	77401			
COPY	33401			
DBUGPT	77003			(DBUG for paper tape)
DDCOPY	77003			(note #3)
FORMAT	33401			
INSTALL	33401			

=========> CONTINUED ON NEXT PAGE.

notes:

#1. LOAD as RTC (without $) if $MMUX and MIGHTY MUX are on the system.
#2. Reload a second copy of the LPT driver for your system as $LPT.
#3. LOAD DISCUTILITY if controller is POINT 4 LOTUS 700
#4. LOAD as DEC (without $) if a Point 4 Mark 8 CPU is being used.
#5. LOAD as MK8 (without $) if a Point 4 Mark 8 CPU is not being used.

CONTINUED FROM PREVIOUS PAGE

KILL	33401			
L.LCMDEFS	77030			
LCMACTIVATE	33401			
LCMMODIFY	33401			
MAIL	33401			
MESSAGES	77001			
PORT	33401			
PROTECT	33401			
O.DMSTB	77003			
QUERY	33401			
RADOC	77003			
RADSC	77003			
RCDTC	77003			
REHASH	77401			
REMOVE	33401			
SAVE	33401			
SHUTDOWN	33403			
VERIFY	33401			
LIBR	33401			
ASSEMBLE	33401			(Filename ASM optional)
EDIT	33401			
RUNMAT	33402			
BASIC	33702			
RUN	33602			

==========> Use DSP to make any necessary modifications to the CONFIG file and to set all multiplexer and peripheral driver attributes as required.
Hit CTRL C to return to SCOPE and run SHUTDOWN.
Then, do a regular IPL to enable the CONFIG file and drivers.
You will not be able to continue the SysGen unless the CARRY light is flashing exactly once per second after you log on. (This indicates that the system clock is going.)
Enter DSP with X as the password, type FDISCSUBS c/r

(c/r = return key), and then load the following with the R command:

DISCSUBS #2	-			
DISCSUBS #3	-			

==========> Press CTRL C to return to SCOPE and run SHUTDOWN.
Do a reglar IPL. Then enter BASIC and press c/r twice to link up BASIC, RUN, and RUNMAT.
(For R4, follow this by typing CTRL C RUNMAT c/r.)
Type SIZE to check the size of the BASIC program area.
Then load the following:
==========> (for text files: COPY filename*T=$PTR)

DEFS	TEXT			
PZ	TEXT			
RMDTC	TEXT			
SYMBOLS	TEXT			
LCMPARAMS	TEXT			

==========> (For BASIC files: enter BASIC, type NEW c/r,
LOAD $PTR c/r, CRTL C and SAVE filename.)

ACCOUNTLIST	BASIC	_____	_____	_____
ACCRESTORE	BASIC	_____	_____	_____
ALOAD	BASIC	_____	_____	_____
BASICTEST	BASIC	_____	_____	(Also DUMP to BT.)_____
BUILDXF	BASIC	_____	_____	_____
EXERCISER	BASIC	_____	_____	_____
EXTRAPORT	BASIC	_____	_____	_____
GUIDE	BASIC	_____	_____	_____
GUIDE.LPT	BASIC	_____	_____	_____
GUIDE.LU	BASIC	_____	_____	_____
GUIDE.BLOCKCPY	BASIC	_____	_____	_____
LCMCHECK	BASIC	_____	_____	_____
LCMCONFIGURE	BASIC	_____	_____	_____
LPTDIAG	BASIC	_____	_____	_____
LPTDIAG2	BASIC	_____	_____	_____
LPTDIAG3	BASIC	_____	_____	_____
LPTDIAG4	BASIC	_____	_____	_____
MTUTL	BASIC	_____	_____	_____
RETRY	BASIC	_____	_____	_____
SETTIME	BASIC	_____	_____	_____
TAPETESTSW	BASIC	_____	_____	_____
UTILITY	BASIC	_____	_____	_____

==========> (For these stand alone binary files: COPY filename*A=$PTR)

DISCDIAG	77003	_____	_____	(obtain from controller vendor)
DISCFORMAT	77003	_____	_____	(obtain from controller vendor)
MUX300DP	77003	_____	_____	_____
MUX310DP	77003	_____	_____	_____

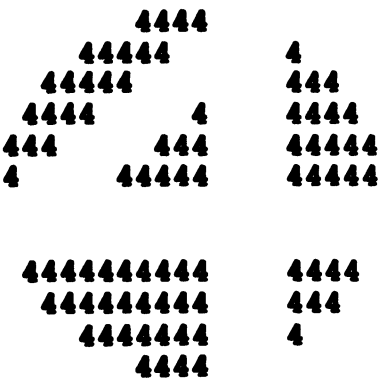
========> FORMAT the USERID file as D2,S14, then use UTILITY (see "Account Utility Program" in the IRIS Manager Reference Manual) to set up the ACCOUNTS as desired.

*** NOW BACKUP THE SYSTEM PACK ***

MANUALS RECEIVED

Quan.	Revision	Date	Description
_____	1017-___	_____	IRIS User Reference Manual
_____	1018-___	_____	IRIS Manager Reference Manual
_____	1015-___	_____	Mighty Mux User's Manual
_____	1028-___	_____	EDSI Multiplexer Diagnostics
_____	_____-___	_____	_____
_____	_____-___	_____	_____
_____	_____-___	_____	_____
_____	_____-___	_____	_____
_____	_____-___	_____	_____
_____	_____-___	_____	_____
_____	_____-___	_____	_____
_____	_____-___	_____	_____

POINT 4 Data Corporation



TECHNICAL MEMORANDUM

TO: ALL IRIS USERS

FROM: IRIS CUSTOMER SUPPORT

DATE: February 3, 1982

SUBJ: INSTALLATION & USE OF IRIS
(7.3 and later revisions)

1. Enclosed is a copy of the new SysGen Log to be used when you SysGen IRIS. Please note that this differs from (supersedes) the procedure described in Section 2 of the IRIS Manager Reference Manual.

2. After an IRIS SysGen, there is a special procedure necessary to use Logical Units from previous IRIS versions. Immediately, after installing the unit under IRIS for the first time, REHASH must be run. REHASH will ask for Logical Unit number to be rehashed. If you do many file creates and deletes on any Logical Unit under IRIS, REHASH can be run periodically to obtain better performance.

3. Because of the use of a buffer pool, it is mandatory that SHUTDOWN be run any time there is to be an interruption of normal IRIS timesharing for any reason. NEVER just hit STOP on the computer.

4. Under IRIS, any halt without a SHUTDOWN is a serious problem: the buffer pool may still contain data which needs to be written to disc. We have provided a manual routine which will allow you to flush the buffer pool and correctly update your disc in case the system goes down. This routine is completely described in Appendix A, "Manual Buffer Pool Flush."

APPENDIX A

MANUAL BUFFER POOL FLUSH

Under IRIS, if the RUN light goes out for any reason other than a SHUTDOWN use this procedure to flush the buffers.

1. Check that the following octal contents are correct in memory:

Location	Octal Contents
7576	6705
7577	6266
7600	102520
7601	60377
7602	102400
7603	42773
7604	102000
7605	62077
7606	6771
7607	63077

2. If the contents are correct, RESET and START at 7600.

3. If the computer now halts with 63077, at location 7607 or 7610, then the manual buffer pool flush has completed successfully. The system is now ready to re-IPL.

4. If the contents of the above locations are not correct, or if the system does not reach a correct 63077 halt, then data may have been irrecoverably lost. Any file accessed since the last IPL may end up randomly missing some of the information which was supposed to have been written to it. You may have to depend on your most recent backups to restore data files to a known good condition.

APPENDIX B
CLEANUP

1. As the files are cleaned up, the phase # and filenames are printed on the terminal. This provides valuable information should problems arise. To suppress the printing of the filenames change location 200 in CLEANUP to a ZERO.

2. After cleaning up Logical Unit 0, an automatic IPL will not occur. Instead, the following message will be printed and the system will halt with a 63077 in the data lights:

"END OF CLEANUP-WHEN SYSTEM HALTS, RE-IPL"

APPENDIX C

TERMINAL CONTROL USER INSTRUCTIONS

1. BASIC I/O: The user may incorporate various terminal control functions into his PRINT, PRINT #, and INPUT statements. In some instances, alternative constructions are given for the control functions. The user should consult the documentation on his particular PORT TYPE to determine which constructions he may use.

1.1 Control of Video Attributes

The user may include various codes (see Section 1.9) in his PRINT and PRINT # statements to control the action of his terminal. For example, to print a blinking error message (on a CRT capable of blinking) the user might do:

```
PRINT 'BB'"ERROR  !"'EB'
```

and the message within the double quotes would be printed and would blink. More than one two-letter code may be included within a single pair of single quotes.

1.2 Protected Fields and Format Mode

One feature of some CRT's is the "protected field." These are sections of the CRT screen specified by the user with 'BP' and 'EP' codes to be write protected. This means that any attempt by an operator to type over these fields will be unsuccessful. Often the user has the option of selectively clearing only the unprotected sections of a screen with a 'CU' code. To make the protected fields operational, the terminal must first be set in "format mode" by sending an 'FM' code to the terminal in a PRINT statement.

1.3 Cursor Positioning For Output

The user may have a statement of the form:

```
PRINT   @X,Y; list;  ⟨@X2, Y2;  list2 . . .⟩
```[1]

where X is an arithmetic expression representing the column number to position to starting with zero and increasing towards the right (as with the TAB function), Y is an arithmetic expression representing the row number starting with zero and increasing towards the bottom, and "list" is any expression list. As many positionings as desired can be implemented in one statement. If Y is omitted, the cursor will be positioned to column X on the current line. The ";" terminating the list may be omitted if desired.

---

1 ⟨. . .⟩ signifies enclosed items are optional.

1.4 Alternative Method For Controlling Video Attributes

On some CRT's (the Beehive B800 is the only CRT in this category at this time), the control of video attributes is done with a four-field command as follows:

```
PRINT @X,Y,T,N; . . .
```

where X and Y are positions on the screen as previously defined and where T represents the attributes to be set in a field N characters long beginning at X,Y. The value of T may be constructed as T = "sum over desired attributes" where the attribute values are 8 for blinking, 4 for reverse video, 2 for dimmed intensity and 1 for write protection. This command doesn't change the position of the cursor.

NOTE: No provision has been made for reading back video attributes that have been set up. This capability, implemented on some CRT's, is used for diagnostic purposes only and must be done at the machine level (i.e., assembly language), not BASIC.

1.5 Cursor Positioning For Input

```
INPUT @X,Y; "prompt" list
```

will position the cursor at X,Y, then output the "prompt" message, and finally accept input into the given "list" of variables. Multiple positionings and prompts are possible in one statement.

1.6 Setting The Length of An Input Field (B800 only)

A three-field construct may be used with the INPUT statement as:

```
INPUT @X,Y,L; . . .
```

which will use the next L characters following the position X,Y for input into the next variable of the statement. It is up to the CRT to generate an EOM code to terminate the input.

1.7 Reading Cursor Position

```
INPUT 'RD' X, Y
```

will return the current cursor coordinates into the variables X and Y. If the variables X or Y are missing, a syntax error will be given at runtime.

1.8 Block input from CRT: (not implemented until nodal input buffer)

```
INPUT 'BT' A$
```

will input into A$ from CRT memory, from current cursor position to next ETX code. Note that the user's I/O buffer must be large enough to hold the entire A$ at one time. Also, if there are any EOM codes (typically Carriage Return, octal 015) in the input string, they will terminate the data put into A$. The rest of the input may then be read with subsequent INPUT statements from the I/O buffer (without using the 'BT').

1.9 TERMINAL CONTROL CODES

Following is a list of all control codes that are planned to be implemented on some terminals. Those marked with a "*" are in a standard minimal subset that have a reasonable expectation of working on any CRT-type terminal for which the system has been implemented. Any others may be implemented for certain terminals which have such capabilities.

@	*	position cursor
RD		read cursor position
CS	*	clear screen
CU		clear unprotected fields
CL		clear from cursor to end of line (unprotected)
CE		clear from cursor to end of screen (unprotected)
MH	*	move cursor home
MU	*	move cursor up
MD	*	move cursor down
ML	*	move cursor left
MR	*	move cursor right
LF		line feed
CR		carriage return, (includes linefeed)
VT		vertical tab
FF		form feed
RB	*	ring bell
BB		begin blink
EB		end blink
BR		begin reverse video
ER		end reverse video
BD		begin dimmed intensity
ED		end dimming
BP		begin write protect
EP		end write protect
BU		begin underline
EU		end underline

1.9 TERMINAL CONTROL CODES (CONTINUED)

BX	begin expanded print
EX	end expanded print
FM	enter format mode (enable write protect)
FX	exit format mode (disable write protect)
LK	lock keyboard
UK	unlock keyboard
BT	begin transmission from CRT memory
ET	ETX code terminating CRT transmission
MP	use memory pointer instead of cursor in next positioning

*Included in standard minimal subset

2. BASIC String Assignment

LET A$(A1,A2) = B$(B1,B2) TO T$(T1,T2):B {,C$. . .}[1]

moves the designated part of B$ into A$ (subscripts optional) until either string ends or until the terminating character T$(T1,T1) is encountered. The optional variable B is set to 0 if T$(T1,T1) was not encountered, or to the byte position in B$ following T$(T1,T1) if it was. (Note that this may = B2+1). Additional string elements (including "TO" forms) may follow, separated by commas.

2.1 The user may now put an integer n between backslashes into literals where 0 n 200 without having the most significant bit (msb) set automatically. If n=0 however, the msb is still set automatically.

3. PORT: To select a desired terminal-dependent translation routine, the user types,

PORT TYPE n

where n is the code for the desired terminal type. The manager must have activiated the corresponding terminal driver by giving it a name starting with "$" so that SIR brings it into core. Otherwise an error message is given.

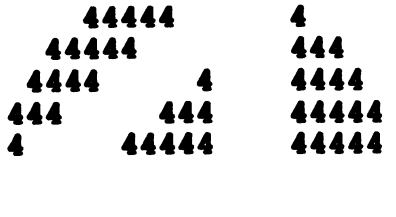
PORT TYPE	n
LSI ADM-3A	3
TERMINET	5

JC:SCM:es

---

1 {. . .} signifies enclosed items are optional.

POINT 4 Data Corporation



TECHNICAL MEMORANDUM

TO: ALL IRIS USERS

FROM: IRIS CUSTOMER SUPPORT

DATE: February 3, 1982

SUBJ: IRIS MODIFICATION RELEASE 1
(7.3 and later revisions)

Your IRIS system has several products which are not described in the IRIS Manager or User Manuals. They are:

1. "Universal" CONFIG file. There used to be a different CONFIG for each type of disc and combining several discs on the same system was a major undertaking. That is all past! The new CONFIG is one standard product for all the discs that we support. Along with your CONFIG file is included a manual caled the "CONFIG LAYOUT," giving all the important parameters for every disc IRIS supports. There is also a BASIC program called GUIDE. When you run the option for logical units, it shows you all the DSP commands you need to change or add to your disc configuration.

2. "Universal" BLOCKCOPY. Systems programmers will find this a valuable utility since it can copy selected portions of any disc to any other kind of disc. Again, GUIDE will describe BLOCKCOPY more fully and show you how to set it up. This is not the "Universal - interactive - failsafe" DDCOPY, but the definite precursor of it.

3. "Universal" $LPT. One line printer driver LPTM handles all printers going through the POINT 4 MIGHTY MUX. Again, the GUIDE program will show you every specific DSP command you need to set up most line printers. This driver also handles VFU, plot mode, back spacing, etc., by a new feature. For example,

```
5  OPEN  #0,  "$LPT"
10  PRINT  #0,1; "\216\\37\\207\"
```

   Record 1 means pass all 8 bits as data, not as text (i.e., no auto LF after CR). Since BASIC now allows you to zero the eighth bit, then any possible 8-bit sequence can now be sent from a BASIC program to your printer. Consult your specific hardware manuals on how to use the VFU, plot mode, etc. LPTP will be the universal printer driver through programmed I/O on device code 17. LPTD will be through the DG 4060 MUX. They will have the same options available as does LPTM.

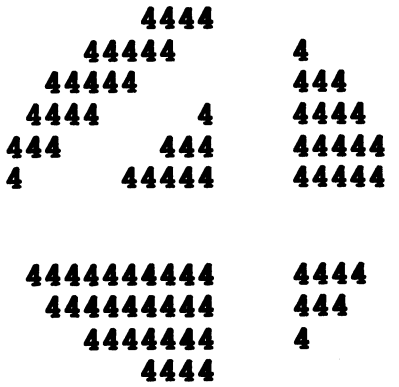
4. Also, for those users with semiconductor memory who felt uneasy about the IRIS buffer pool in core, we have added an optional CONFIG file setting which is described in the enclosed addenda to our 7.3 Manuals. With the setting called NDP, any extra danger of buffer pool is completely eliminated and much of the benefit still remains.

5. The limitation that, if you use the MCT, then it must be the swapping disc, has been removed.

6. NOTE: MBUS is now 20600. Please set this value in INFO in CONFIG for all 7.3 systems.

7. Processors and drivers issued before the release of IRIS 7.3 are different. Check that you do not load the wrong revision level by mistake.

POINT 4 Data Corporation



T E C H N I C A L
M E M O R A N D U M

TO: ALL IRIS USERS

FROM: IRIS CUSTOMER SUPPORT

DATE: February 3, 1982

SUBJ: IRIS MODIFICATION RELEASE 2
(7.3 and later revisions)

A) SCO Numbering
B) EXERCISER
C) Buffer Pool
D) Status of IRIS

A. SCO NUMBERING

First, what is an SCO? At POINT 4 our customer support department helps our users solve any problems they may encounter using IRIS. Periodically, we send these solutions to all the IRIS users currently on subscription service. This collection of patches is carefully selected, tested, and released as an SCO ("software change order"). Each patch sheet has a description of the problem it solves, and so some users are tempted to pick and choose among the updates and only enter part of the SCO. Please enter the complete SCO because:

1. It is hard for us to help you solve new problems if you are not running our standard software with all released patches.

2. Each succeeding SCO will assume that the previous SCO is already entered.

B. EXERCISER

A new BASIC program is included as both papertape and listing. It is a very simple, convenient test of CPU, core, and disc which can be run while other users are running. It will print instructions when you run it. EXERCISER will run forever, until aborted by <CTRL-C> or until it finds any error. It cannot pin-point the problem as CPU, core, or disc, but it can detect subtle (but serious), intermittent hardware problems at a very early stage. POINT 4 recommends you leave it running regularly overnight and over weekends.

C. BUFFER POOL

The following discussion is included to help you understand the buffer pool, NDPF and the trade-off between performance and security.

EXTRANEOUS DISC READS: Often a record is read and updated several times. If the record is already in a disc block in the buffer pool, then each read gets the core copy and does not have to do a disc access. Because of the buffer pool, extraneous disc reads are eliminated and performance improves considerably.

EXTRANEOUS DISC WRITES: Ideally, extraneous disc writes can be eliminated by updating just the core copy in the buffer pool so that the block is only written once after all updates are done. Disc access is reduced and performance improved.

DIRTY PAGE: Any block which has been updated in core but not written to disc is a dirty page. The system flushes dirty pages to disc if it is idle. However, if it is busy, the most used blocks may stay dirty for several hours or more.

MANUAL BUFFER POOL FLUSH: This was described in the Installation & Use of IRIS Memo, Appendix A (see Section 8). It writes all dirty pages to disc, assuming core is intact. If the manual flush does not get a good halt, there is no way to predict what information was lost. Even operations completed several hours ago may be incomplete on disc. Losing the buffer pool with NDPF=0 is a serious problem. Files may have to be restored from a backup and all the updates reentered.

NDPF (No Dirty Page Flag): Cell 606 in CONFIG (bit 15).
0 in NDPF means the flag is not set. This gives maximum speed performance as all extraneous reads and writes are eliminated. If there is a system crash, the buffer pool must be flushed manually.

1 in NDPF sets the flag (i.e., put 100000 in 606). This forces all writes to disc and uses the buffer pool to improve system performance on disc reads. Since extraneous disc reads are eliminated but not extraneous disc writes, about one-half of the speed advantage of the buffer pool is gained (if there are an equal number of reads and writes) without losing any security.

D. STATUS OF IRIS

The following areas are still unresolved in IRIS:

1. DSP breakpoints are not functional.

2. DSP Paper Tape punch commands are not functional.

3. DSP Search with a terminating address at 177777. The value 177776 should be used instead of 177777.

4. COPY $PTP = $PTR does not work correctly.

```
10 DIM L$[20]
20 DIM A$[100],B$[100]
30 LET A$="HAVE COMPLETED THROUGH BLOCK ###### OF WRITE OUT PHASE "
40 LET B$="HAVE COMPLETED THROUGH BLOCK ###### OF READ BACK AND VERIFY PHASE"
50 DIM 4%,A[63],B[63]
60 PRINT "\215\THIS CORE AND DISC EXERCISER PROGRAM WAS NOT DESIGNED TO BE A"
70 PRINT "REPLACEMENT FOR A COMPREHENSIVE STAND-ALONE RELIABILITY PROGRAM. "
80 PRINT "RATHER, IT IS A CONVENIENT TOOL WHICH CAN BE RUN USING LIVE"
90 PRINT "DATA PACKS, WITHOUT HAVING TO ASK EVERYONE ELSE TO LOG OFF. "
100 PRINT "EDS ADVISES YOU RUN IT OVERNIGHT OR OVER WEEKENDS.  ANY ERROR"
110 PRINT "FOUND IN CORE OR ON DISC WILL ABORT THE PROGRAM AND PRINT"
120 PRINT "AN ERROR MESSAGE. "
130 LET N=1
140 PRINT "\215\\215\"
150 PRINT "CHOOSE THE # OF BLOCKS TO BE USED BY THIS EXERCISER. "
160 PRINT "THE # SHOULD BE GREATER THAN THE # OF BLOCKS IN THE BUFFER"
170 PRINT "POOL, IF POSSIBLE.  (TO DETERMINE # OF BLOCKS IN THE BUFFER"
180 PRINT "POOL, USE DSP TO DUMP LOCATION 664 IN CORE. REF: 7.3 MANAGER"
190 PRINT "MANUAL PG. A5-2. CONVERT THE # FOUND THERE FROM OCTAL TO DECIMAL. )"
200 PRINT "MAKE SURE THAT THE CHOSEN BLOCK COUNT DOES NOT EXCEED THE # OF"
210 PRINT "BLOCKS AVAILABLE TO THIS ACCOUNT ON THE SELECTED LOGICAL UNIT. "
220 INPUT "ENTER # OF BLOCKS TO USE : "C
230 PRINT
240 INPUT "ENTER LOGICAL UNIT # TO USE : "L
250 LET L$=L
260 LET L$=L$[2, LEN L$-1]
280 PRINT "\215\\215\\215\"
290 SIGNAL 3,0
295 LET L$=L$,"/EXERCISERFILE!"
300 BUILD #0,L$
310 MAT  WRITE #0,0;A
320 CLOSE #0
330 OPEN #0,L$
340 PRINT "\215\"
350 FOR J=1 TO C
360   LET P=J*.98765432101234*N
370   FOR I=0 TO 63
380     LET A[I]=P
390   NEXT I
400   IF  FRA (J/15)<>0 GOTO  430
410   IF N<>1 SIGNAL 3,40
420   PRINT  USING A$;J ! TO OVERLAP EDS MUX VS DISC
430   MAT  WRITE #0,J;A
440 NEXT J
450 FOR K=1 TO C
460   LET M=K*.98765432101234*N
470   MAT  READ #0,K;B
480   FOR I=0 TO 63
490     IF B[I]<>M GOTO 630
500   NEXT I
510   IF  FRA (K/15)<>0 GOTO  540
520   IF N<>1 SIGNAL 3,40
530   PRINT  USING B$;K ! TO OVERLAP EDS MUX VS DISC
540 NEXT K
550 PRINT "\215\\215\"
560 PRINT "EDS CORE AND DISC EXERCISER (VERSION 1)"
570 PRINT "WITH # OF BLOCKS IN TEST = "C
580 PRINT "PASS #",N,"COMPLETED OK.  (NO ERRORS)"
590 PRINT "PRESS CONTROL C TO ABORT.  THEN KILL EXERCISERFILE. "
600 LET N=N+1
610 SIGNAL 3,100
620 GOTO  340
630 PRINT "\215\\215\ERROR!  DATA READ BACK FROM DISC IS NOT WHAT WAS WRITTEN.
640 PRINT "\215\THIS IS A MAJOR HARDWARE FLAW.  EDS RECOMMENDS YOU SUSPEND"
650 PRINT "ALL CRITICAL SYSTEM USAGE UNTIL CPU, MEMORY, OR DISC DIAGNOSTICS"
660 PRINT "HAVE LOCALIZED THE PROBLEM AND THAT COMPONENT IS FIXED. "
```

```
**************************************
*                                    *
*                                    *
*         EXERCISER PROGRAM          *
*                FOR                 *
*           DISC  &  CORE            *
*                                    *
**************************************
```

POINT 4 DATA CORPORATION
2569 McCabe Way / Irvine, California 92714 (714) 754-4114 / TWX (910) 595-1113

December 30, 1981

T E C H N I C A L M E M O

SUBJ: DBUG

The 7.5 DBUG is CTU oriented and will accept the CTU commands (i.e., Z) but will not accept papertape reader commands (i.e., R1).

To read papertapes, IPL and shutdown to DBUGPT. Then put 25000 in the switches and do a reset-start. On a POINT 4 after pressing stop, do a J25000.

TECHNICAL MEMO

March 31, 1978

TO : **All IRIS Users**
FROM : Leon Remus
SUBJECT: Using EXTRAPORT

EXTRAPORT is a BASIC program that assigns other programs to run on Phantom Ports if you are logged onto the Manager account (see User's Manual pg. 1-10 "How to Use a Phantom Port").

To use this program, type in EXTRAPORT. Your terminal will respond with "!" or "ALL PHANTOM PORTS ARE BUSY ! ! !". If all ports are busy, it is best to wait for a few minutes and try again. If the response was "!", then input the desired command.

Example:

```
#EXTRAPORT (return)
!LIBR @^[$LPT] (return)
```

The program will find a vacant Phantom Port and send the command string to it. Then, EXTRAPORT will check the status of the port that was selected and will respond with a message accordingly.

Example:

```
LIBR IS RUNNING ON PORT 1
        or
LIBR FAILED ! ! !
```

If the command fails, this means there was some error in the command string. To see the error message, enter the same command directly on your terminal.

All Phantom Ports on your IRIS system are automatically found by EXTRAPORT so there is no need to change any constant in EXTRAPORT before running this program on any IRIS R7.3 system.

7:81:MMH:S104
POINT 4 Data Corporation

T E C H N I C A L M E M O R A N D U M

TO: All IRIS Users

FROM: Max M. Heller
Vice President, Systems Support

DATE: July 29, 1981

SUBJ: XREF (Cross-Reference Utility for BASIC Programs)

I. The XREF system consists of a set of nine programs written in Business BASIC. They read specified text file versions of BASIC programs and produce a selected number of paginated listings on a designated system line printer. The output consists of:

1. Title page - name of recipient of listing(s).

2. The text file - line numbers referenced by a GOSUB or GOTO statement are prefixed with a "+".

3. A symbol table - BASIC variables in alphabetic sequence with associated program line number references - in many cases, the line number is tagged to indicate usage.

4. Channel numbers - with associated line number references - followed by BASIC statements with their associated line number references.

5. Program line numbers used as targets - in ascending numeric sequence with associated targeting line numbers - targeting line numbers using GOSUB are suffixed with an "*".

6. A legend of cross-reference symbols - a glossary of symbols used by XREF.

A listing for XREF5. produced by XREF is enclosed as a sample report.

II. Following are the text file names and their associated BASIC program names for the XREF system. They should be loaded into BASIC and saved on logical unit 5 as follows:

Text File	Save As
XREF.	XREF
XREF1.	XREF1
XREF2.	XREF2
XREF3.	XREF3
XREF4.	XREF4
XREF5.	XREF5
XREF6.	XREF6
XREFA.	XREFA
XREFB.	XREFB

III. The system is entered through XREF at the system prompt. It chains among modules as required. A flowchart of the XREF system is attached.

The following is the first screen displayed:

```
PORT NUMBER:  18      CROSS-REFERENCE SELECT        XREF 2.0 05/13/81


                FILE(S) PREFIX:  ______________
                  LOGICAL UNIT:  __
                  LINE PRINTER:  _


COMMENT:  CREATE NEW WORK FILE? (Y/N)
COMMAND:
MESSAGE:
```

If the response is "N", the system requests "COPY FROM SAVED WORK FILE? (Y/N)". If the response is again "N", the system requests entry of the line printer #. A response of 1, 2 or a null return is acceptable. The null return will later open the $LPT driver; the 1 or 2 will use $LPT1 or $LPT2 respectively.

If the file has already been processed the following screen display will appear:

```
PORT NUMBER:  18      CROSS-REFERENCE SELECT       XREF1 2.0 05/13/81


--> NO FILES SELECTED


COMMENT:  'XREF' COMPLETE, MORE PROGRAMS TO PROCESS (Y/N) ?
COMMAND:
MESSAGE:
```

and the XREF should be terminated and restarted to build a new work file as described below.

If the work file has not been processed, a screen similar to the following will appear, and processing can continue.

```
PORT NUMBER: 18       CROSS-REFERENCE SELECT       XREF1 2.0 05/13/81


LOGICAL UNIT #3

NO.   *  FILE NAME
===   =  =========
01.   T  XREF5.



COMMENT:  A=ALL, E=EXECUTE, P=PAGE, R=RESTART, W=WARP, ?=HELP, AND
          nn=FILE NUMBER
COMMAND:
MESSAGE:
```

If the response to the first screen,

is "N", but the response to the next query,

```
    COPY FROM SAVED WORK FILE? (Y/N)
```

is "Y", the system first requests input of the line printer number as previously described, then copies a new work file from one previously saved displaying the following:

```
PORT #:  18           COPY EDITSV0nn WORK FILE     XREFB 2.0 07/16/81


#COPY (00) EDITSV0nn!=EDITSV0nn.SAVE
```

and then displays the file list for further processing.

If the reponse to the first screen

```
CREATE NEW FILE? (Y/N)
```

is "Y", the system offers two choices as follows:

```
CREATE 'LIBR' WORK FILE? (Y/N)
```

A response of "Y" positions the cursor for entry of the file(s) prefix, logical unit number and line printer number successively. (Entry of a "?" at any of these inputs will display a "help" screen.) The system will then chain to a sorted "LIBR" to create the work file (EDITSV0xx, xx=port #) and then chains to copy the work file to "EDITSV0xx.SAVE" for later access as required. After the copy, the system chains to the edit module and displays the contents of the work file for further processing.

If the response to

```
CREATE 'LIBR' WORK FILE? (Y/N)
```

is "N", the system prompts for entry of the logical unit number and then positions for direct entry of text file names. After all the file names have been entered, a prompt message requests entry of the line printer number and then chains to copy the work file (EDITSV0xx) to the saved file (EDITSV0xx.SAVE). When the copy is completed, the system chains to the edit module and displays the work file for further processing.

After entering the number of XREF listings for each text file to be processed, the system prompts for entry of the recipient's name.

After these entries are made, there is a help module available. Its access is described in a displayed message during the work file edit procedure.

Upon completion of editing, the (E) execute input by the user will prompt

```
RUN ON PHANTOM PORT? (Y/N)
```

A "Y" response will enable a phantom port and transfer further processing there. The system will then print

```
RUNNING ON PORT p (p=phantom port #) and chain
to SCOPE
```

If the response is "N", the processing of XREF will continue on the user's port and when complete, will chain to SCOPE.

IV. All references to XREF programs for chaining within the XREF system are hard coded for Logical Unit #5. The program names and the line numbers within each program where such references appear are as follows:

```
                        XREF
317 LET I$="RUN 5/XREFA"
966 LET I$="5/XREFB\215\"
1955 LET I$=I$,"5/XREF1"

                        XREF1
640 IF I$="Y" LET F$= "RUN 5/XREF"
2498 LET F$="RUN 5/XREF2"

                        XREF2
890 LET F$="5/XREF3"

                        XREF3
960 CHAIN "RUN 5/XREF4"

                        XREF4
670 CHAIN "RUN 5/XREF5"

                        XREF5
660 CHAIN "RUN 5/XREF6"

                        XREF6
850 CHAIN "5/XREF2"

                        XREFA
930 LET C$=C$,"5/XREF1"

                        XREFB
240 LET C$="COPY <00>",F$,"!=",F$,".SAVE\215\5/XREF1"
```

All references to phantom ports in XREF programs use ports numbered 1 through 4. The program names and the line numbers where such references appear are as follows:

```
                        XREF1
5000 LET P1=1 ! This is first phantom port number
5010 LET P2=4 ! This is last phantom port number


                        XREF2
920 IF SPC(6)>=1  IF SPC(6)<=4  LET F$="0/BYE"
```

The COPYRIGHT line appearing at the bottom of each page of the XREF listings is written by two lines in each of two programs. The second program line in each should be deleted or modified to remove " by POINT 4 DATA CORPORATION" from your listings. The first program line in each prints the copyright notice. Both lines in each program can be deleted if no copyright notice is desired.

XREF4

1010
1020

XREF5

950
960

XREF is supplied without charge and without support. Listings can be obtained by dumping the text files indicated in the list above or by using XREF.

MMH/st
Enclosures: XREF flowchart
XREF5. listing

```
  1 REM                    XREF5
  2 REM               COPYRIGHT 1981 BY POINT 4 DATA CORPORATION
 10 REM  PROGRAM NAME = "XREF5"
 10 REM
 10 REM  THIS PROGRAM IS COURTESY OF MAX HELLER, P4DC, AND HIGHLY
 10 REM  MODIFIED DURING THE LAST SEVERAL YEARS BY DON DOBBS.
 10 REM
 10 REM  THIS PROGRAM USES THE INDEX INFORMATION PRODUCED BY
 10 REM  'XREF5' TO FORMAT THE OUTPUT XREF LISTING.
 10 REM
 10 REM  CHNL #1 - INPUT TEXT FILE OF BASIC PROGRAM
 10 REM  CHNL #2 - OUTPUT TEXT FILE FORMATTED FOR $LPT
 10 REM  CHNL #6 - CONTROL FILE OF FILE NAMES
 10 REM               REC #0 - LAST PAGE # FORMATTED (1%);
 10 REM                        TOTAL # COPIES TO BE PRINTED (1%)
 10 REM               REC #1 - NAME OF INPUT TEXT FILE
 10 REM               REC #2 - NAME OF OUTPUT TEXT FILE FOR $LPT
 10 REM               REC #3 - NAME OF INDEX DIRECTORY FILE
 10 REM                        FOR SORTED XREF'S
 10 REM               REC #4 - DATE AND TIME OF PRINTOUT
 10 REM  CHNL #7 - INDEX DIRECTORY FILE NAME
 10 REM
 10 REM  NOTE: CHANNELS ALREADY OPENED BY PREVIOUS PROGRAM.
 10 REM
 10 REM  THIS PROGRAM 'CHAINED TO' FROM "XREF4"
 10 REM
 10 REM
 10 IF ERR 0 GOTO 1140
 20 DIM A$[20],A1$[5],D$[22],F$[1],F1$[1]
 30 DIM P$[20],S$[1],V$[11],V1$[11]
 40 DIM 1%,I,I1,I2,I3,L1,L2,L3,P,T,W9,X,X1,X2,2%
 50 LET T=1    !  DEFAULT LEFT MARGIN
 60 LET W9=79  !  WIDTH FROM TAB T TO PAGE PERF. DOT (.)
 70 LET X1=7   !  WIDTH FOR EACH XREF PRINTING
 80 LET X2=11  !  TAB OF FIRST LINE # CROSS-REF
 90 LET V1$=""!  INITIAL SEARCH KEY
100 READ #6,4;D$;
110 READ #6,1;A$;
120 READ #6,2;P$;
130 READ #6,0;P;
140 PRINT @0,10;"CROSS REFERENCE FORMATTING PHASE ";
150 SIGNAL 3,0
160 GOSUB 800
170 LET I2= INT ((W9-10)/X1)  ! CALCULATE # XREFS TO PRINT ON EACH LINE
180 LET L2=60 !    # PRINT LINES PER PAGE
+190 LET I1=1
200 LET X=T+X2 !  INIT TAB SETTING FOR 1ST XREF #
+210 FOR I=I1 TO I2
220   SEARCH #7,3,1;V1$,V1,V2
230   IF V2=2 GOTO 620
240   IF V2<>0 STOP
```

```
 250    LET V$=V1$
 260    IF V1$[1,1]="\207\" LET V$[1,5]=V1$[2,4]," "
 270    IF V1$[1,1]<>"\207\" IF F1$="\207\" GOSUB 1110
 280    LET F1$=V1$
 290    IF V1$[1,1]="\376\" LET V$[1,5]=V1$[2,5]," "
 300    IF V1$[1,1]="\376\" IF F$<>"\376\" GOSUB 1110
 310    LET F$=V1$
 320    IF A1$=V$[1,5] GOTO 470
 330    LET A1$=V$[1,5]
 340    IF X<>T+X2 GOSUB 880
 350    IF L1>L2 GOTO 380
 360    GOSUB 880
 370    IF L1<=L2 GOTO 400
+380    GOSUB 790
 390    GOSUB 880
+400    LET X=T+X2
 410    PRINT #2; TAB T;A1$;
 420    GOSUB 980
 430    LET L3=0
 440    GOSUB 680
 450    LET I1=2
 460    GOTO 210
+470    IF L3=0 GOTO 540
 480    PRINT #2; TAB T;A1$;
 490    GOSUB 980
 500    PRINT #2; TAB T+X2-2;"(CONTD)";
 510    LET I=I+1   !MAKE ROOM FOR WORD "(CONTD)" IN TABBING COUNTER
 520    LET X=X+X1
 530    LET L3=0
+540    GOSUB 680
 550 NEXT I
 560 GOSUB 880
 570 IF L1<=L2 GOTO 190
 580 GOSUB 790
 590 LET L3=1
 600 GOSUB 880
 610 GOTO 190

+620 REM  CODING TO WRAP IT UP...
 630 GOSUB 1050
 640 CLOSE #2
 650 OPEN #2,P$ !  POSITION TO START OF FILE
 660 CHAIN "RUN 5/XREF6"

 670 STOP
```

```
+680 REMPG
 680 REM
 680 REM  ** SUBR. TO PRINT A LINE REF. **
 680 LET S$=" "
 690 IF V$[10,10]<>"[" GOTO 760
 700 FOR I3=6 TO 9
 710   IF V$[I3,I3]<>" " GOTO 730
 720 NEXT I3
+730 LET V$[10,10]=" "
 740 IF I3=6 LET S$="["
 750 IF I3>6 LET V$[I3-1,I3-1]="["
+760 PRINT #2; TAB X-1; S$; V$[6,10];
 770 LET X=X+X1
 780 RETURN

+790 REM  ** SUBR. TO PRINT HEADINGS **
 790 PRINT #2; "\214\"
+800 LET P=P+1
 810 LET L1=10
 820 PRINT @45,21; "---> PAGE NUMBER"; P;
 830 SIGNAL 3,0
 840 PRINT #2; "\215\\215\\215\"
+850 PRINT #2; TAB T; D$[1,19]; TAB (T+30); A$; TAB (T+65); "PAGE";
 860 PRINT #2; USING "###"; P; TAB (T+W9); ". \215\"
 870 RETURN

+880 REM  ** SUBR. TO PRINT A BLANK LINE **
 890 PRINT #2; "\215\";
 900 LET L1=L1+1
 910 IF L1>L2 GOSUB 930
 920 RETURN

+930 REM  ** SUBR. TO PRINT COPYRIGHT & DOT (.) **
 930 PRINT #2;  "\215\\215\";
 940 GOSUB 850
 950 PRINT #2;  TAB (T+13); "COPYRIGHT (C) "; D$[9,12];
 960 PRINT #2;  " by POINT 4 DATA CORPORATION"
 970 RETURN

+980 REM  SUBR TO EXPAND SHORTENED KEYS
 980 IF A1$="RANDO" PRINT #2; "M";
 990 IF A1$="RESTO" PRINT #2; "R";
1000 IF A1$="RETUR" PRINT #2; "N";
1010 IF A1$="SEARC" PRINT #2; "H";
1020 IF A1$="SIGNA" PRINT #2; "L";
1030 IF A1$="SUBRO" PRINT #2; "UTINE";
1040 RETURN

+1050 REM  SUBR TO SPACE TO BOTTOM OF PAGE & PRINT FOOTING
1060 FOR I=L1 TO L2
1070   PRINT #2;
```

```
 1080 NEXT I
 1090 GOSUB 930
 1100 RETURN

+1110 REM  SUBR TO ACCOMPLISH A PAGE BREAK
 1110 GOSUB 1050
 1120 GOSUB 790
 1130 RETURN

+1140 IF SPC(8)<>99 GOTO 1170
+1150 PRINT 'CS';"NORMAL EXIT FROM XREF\215\";
 1160 CHAIN ""

+1170 IF ERR 0 GOTO 1150
+1180 PRINT @09,23;'CLRB';"ERROR";SPC(8);"AT LINE";SPC(10);
 1190 SIGNAL 3,50
 1200 GOTO 1180

 9999 END
```

A$	20:	110;	850						
A1$	20:	320	330=	410	480	980	990	1000=	1010
	1020	1030							
D$	20:	100;	850	950					
F$	20:	300	310=						
F1$	20:	270	280=						
I	40:	[210	510	510=	550]	[1060	1080]		
I1	40:	190=	210	450=					
I2	40:	170=	210						
I3	40:	[700	710	720]	740	750			
L1	40:	350	370	570	810=	900	900=	910	1060
L2	40:	180=	350	370	570	910	1060		
L3	40:	430=	470	530=	590=				
P	40:	130;	800	800=	820	860			
P$	30:	120;	650						
S$	30:	680=	740=	760					
T	40:	50=	200	340	400	410	480	500	850
	860	950							
V$	30:	250=	260=	290=	320	330	690	710	730=
	750=	760							
V1	220;								
V1$	30:	90=	220;	250	260	270	280	290	300
	310								
V2	220;	230	240						
W9	40:	60=	170	860					
X	40:	200=	340	400=	520	520=	760	770	770=
X1	40:	70=	170	520	770				

X2 40: 80= 200 340 400 500

#2	410	480	500	640	650	760	790	840	850
	860	890	930	950	960	980	990	1000	1010
	1020	1030	1070						
#6	100	110	120	130					
#7	220								
CHAIN	660	1160							
CLOSE	640								
DIM	20	30	40						
END	9999								
ERR	10	1170							
GOSUB	160	270	300	340	360	380	390	420	440
	490	540	560	580	600	630	910	940	1090
	1110	1120							
OPEN	650								
PRINT	140	410	480	500	760	790	820	840	850
	860	890	930	950	960	980	990	1000	1010
	1020	1030	1070	1150	1180				
READ	100	110	120	130					
RETURN	780	870	920	970	1040	1100	1130		
SEARCH	220								
SIGNAL	150	830	1190						
SPC	1140	1180							
STOP	240	670							
SUBROUTINE	680	790	800	850	880	930	980	1050	1110

190	570	610			
210	460				
380	350				
400	370				
470	320				
540	470				
620	230				
680	440*	540*			
730	710				
760	690				
790	380*	580*	1120*		
800	160*				
850	940*				
880	340*	360*	390*	560*	600*
930	910*	1090*			
980	420*	490*			
1050	630*	1110*			
1110	270*	300*			
1140	10				
1150	1170				
1170	1140				
1180	1200				

LEGEND OF CROSS-REFERENCE SYMBOLS

THE FOLLOWING SYMBOLS MAY APPEAR AS PREFIXES TO THE STATEMENT NUMBERS IN THE SOURCE LISTING.

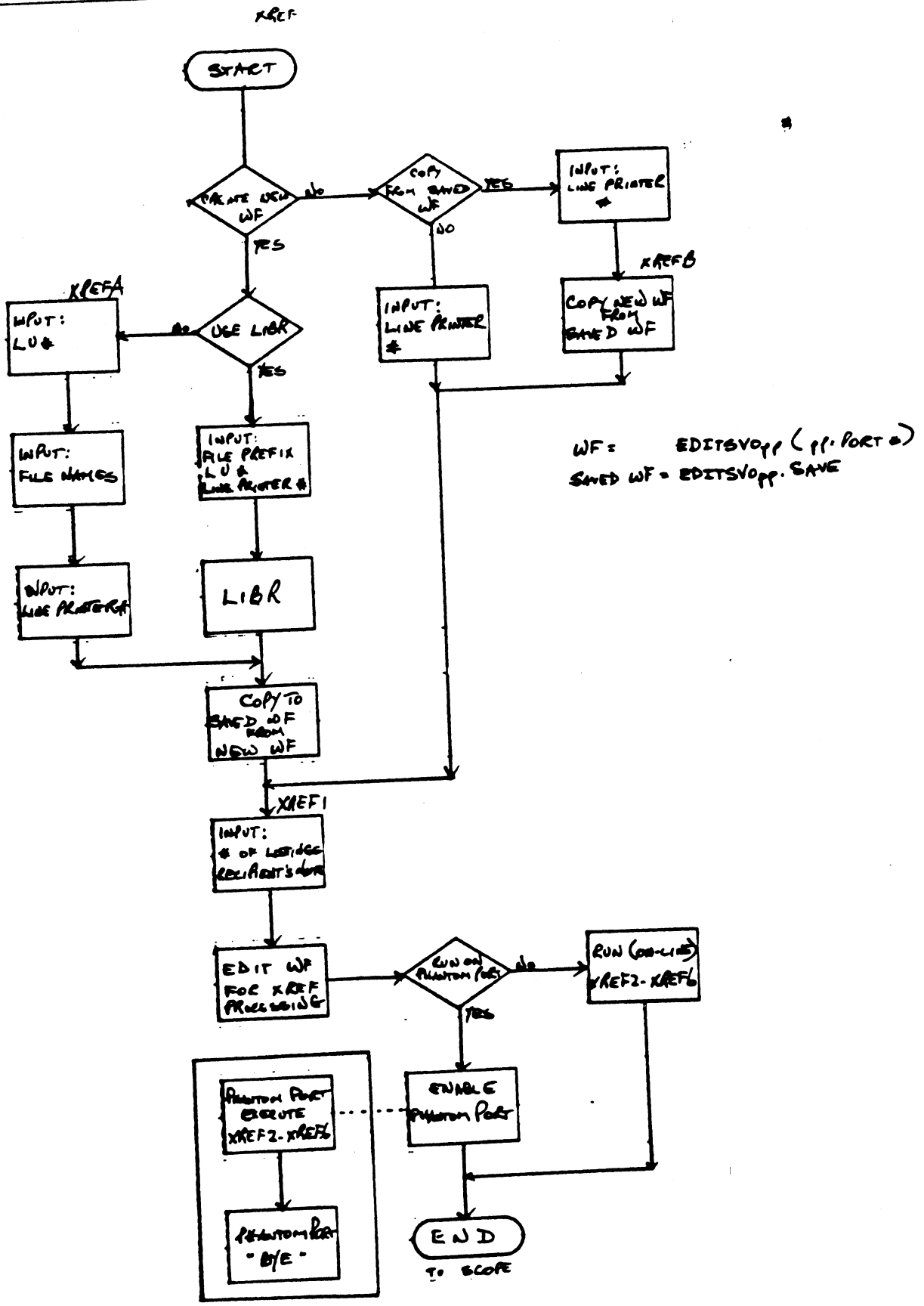
+	This statement is the target of one or more GOTOs or GOSUBs

THE FOLLOWING SYMBOLS MAY APPEAR AS SUFFIXES TO THE CROSS-REFERENCE LINE NUMBERS.

:	Variable is declared in a DIM statement
^	Variable is used in a DIM statement to dimension an array or string variable
=	Variable is the target of an assignment
;	Variable is assigned a value as the result of a READ, INPUT, SEARCH, or CALL statement
[	Variable is used as control in a FOR..NEXT (The start of a FOR..NEXT loop)
]	The close of a FOR..NEXT loop
#	Variable is used to designate I/O channel #
*	The line number reference to this statement is a GOSUB

Programmer: MAX M. HELLER Program No.: XREF SYSTEM Date: 7-23-81 Page: 1/1

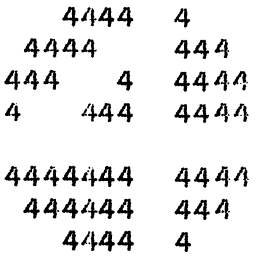
Chart ID: ____ Chart Name: ____ Program Name: ____



WF = EDITSVOpp (pp: PORT #)

SAVED WF = EDITSVOpp.SAVE

POINT 4 Data Corporation



TO: All IRIS Users

FROM: Applications Software Department

DATE: December 4, 1981

SUBJ: ACS.VERIFY utility

The Application Control System VERIFY function is a utility program which verifies the check codes of BASIC programs running under IRIS R7.5. For each system, the filenames and the check codes (when applicable) are stored in a parameter file P.<system>. The program reads the filename from the parameter file and verifies its existence. If the file is not found, the filename and a message "file not found" are logged into the log file. If the file exists and is a BASIC file, the program loads the BASIC file, returns to SCOPE and executes the VERIFY processor. If the check code generated by VERIFY does not match the check code in the parameter file, then the filename, the check code of the parameter file and the check code generated by VERIFY are logged into the log file.

The requirements for the ACS.VERIFY program are:

- The ACS.VERIFY program must be run from the manager account
- The ACS.VERIFY program must have the following files residing on the manager account's assigned logical unit:

ACS.WORK	2 blocks
ACS.VERIFY	12 blocks
ACS.VERIFY2	5 blocks
ACS.VERIFY3	5 blocks
ACS.VERIFY4	8 blocks

- There must be enough blocks allocated to the manager account on the specified logical unit to build the following temporary work files:

PRMTR.IDX	42 blocks
PRMTR.LOG	108 or 22 blocks

- The total manager account's requirments = 182 blocks

The VERIFY program is accessed by entering ACS.VERIFY at the IRIS system prompt (#). The program displays the System Selection Menu. After the number of the selected system is entered, the program verifies that the associated parameter file is on the system. If the parameter file is not found the program responds:

PARAMETER FILE NOT ON SYSTEM - OPEN ERROR

This message is also displayed if the program cannot open the file. If this message is displayed, the program returns to the System Selection Menu. A CR may be entered to terminate the program and to chain to the system prompt.

After a system has been successfully selected, the program responds:

ENTER BASIC FILES LU#:

The program accepts any number between 0 and 99; a CR defaults to the assigned LU#.

The program then asks for the files to be verified by displaying:

1. ALL
2. RANGE
3. LIST

ENTER METHOD OF SELECTING FILES (1-3):

Selection of 1 (ALL) will process all files in the selected system parameter file.

Selection of 2 (RANGE) will process a specified range of files in the selected system parameter file. The program prompts:

FROM FILE PREFIX (CR = FIRST):

and

TO FILE PREFIX (CR = LAST):

The RANGE function allows the user to process one

section at a time or to restart in the middle if the verifying process was interrupted. (Entering CR to both prompts is equivalent to the ALL function.)

Selection of 3 (LIST) will process a user-selected list of one or more files in the selected system parameter file. The program displays:

BUILDING INDEX FILE

The program then asks for the filenames one at a time.

1. ENTER FILENAME (CR TO END):

The user enters the first file to be verified. The program prompts for additional filenames. When all filenames have been entered, a CR completes the list entry. All filenames entered must be in the parameter file and on the disc.

After the file selection phase is completed the program builds an error logging index file and displays:

BUILDING LOG FILE

The program will then go into the verifying loop. If the file is not found or if it encounters a program check code that doesn't match the check code in the parameter file, the program logs that file as an error.

When the program has finished verifying the files, it checks the log file for errors. If the file is empty the program displays:

NO ERRORS HAVE BEEN FOUND
PROCESS COMPLETED

and the program chains back to the System Selection Menu, otherwise, the program prompts:

< n > ERRORS HAVE BEEN LOGGED INTO THE LOG FILE
SELECT OUTPUT: 1=CRT 2=PRINTER 3=DISC (1-3)

Selection of 1 (CRT) will output the report to the screen.

Selection of 2 (PRINTER) will output the report to the specified printer. The program asks:

ENTER PRINTER DEVICE CODE:

The input must begin with a dollar sign ($). If an error occurs it will be reported and the program will repeat the device code question. A CR may be entered

to return to the output selection screen.

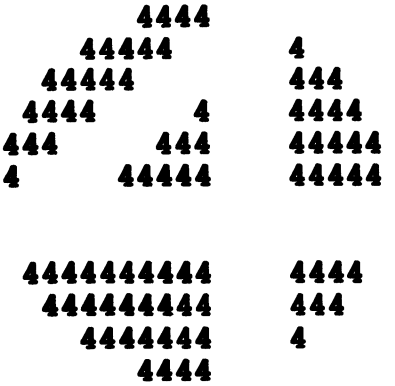
Selection of 3 (DISC) will output the report to the specified text file.

ENTER FILENAME:

The program will try to build the file. If it already exists the user must include an exclamation point (!) at the end of the filename. A CR may be entered to return to the output selection screen.

The log is then output using the selected method and the program chains back to the System Selection Menu. A CR may be entered to chains to the system prompt.

POINT 4 Data Corporation



T E C H N I C A L
M E M O R A N D U M

TO: All FORGE Users

FROM: Applications Software Department

DATE: June 30, 1982

SUBJ: FORGE (FOR ON-LINE REAL-TIME GENERAL EDITING)
LOADING INSTRUCTIONS

The following files are the text file versions for the FORGE text editor. They should be loaded into BASIC and saved on logical unit 5 as program names represented by the text file name without the prefix "T.".

Text File	Saved As
T.FORGE	5/FORGE
T.FORGE1	5/FORGE1
T.FORGE2	5/FORGE2
T.FORGE21	5/FORGE21
T.FORGE22	5/FORGE22
T.FORGE23	5/FORGE23
T.FORGE3	5/FORGE3

If an installation requires that the FORGE text editor reside on a logical unit other than 5, the following program line numbers should be appropriately modified:

```
FORGE
    1460 OPEN #5,"5/FORGE1"
    1940 LET A$="RUN 5/FORGE1"

FORGE1
    2920 IF C=97 CHAIN "RUN 5/FORGE4"
    3010 CHAIN "RUN 5/FORGE"
    5140 IF C=9 CHAIN "RUN 5/FORGE2"
    5170 CHAIN "RUN 5/FORGE3"

FORGE2
    720 CHAIN "RUN 5/FORGE23"
    860 IF A<14 CHAIN "RUN 5/FORGE21"
    870 CHAIN "RUN 5/FORGE22"
    890 CHAIN "RUN 5/FORGE1"
```

```
FORGE21
    110 CHAIN "RUN 5/FORGE1"
    520 CHAIN "RUN 5/FORGE2"
    550 CHAIN "RUN 5/FORGE2"
    1390 IF C$="" CHAIN "RUN 5/FORGE2"

FORGE22
    100 CHAIN "RUN 5/FORGE1"
    500 CHAIN "RUN 5/FORGE2"
    580 IF C$="" CHAIN "RUN 5/FORGE2"

FORGE23
    110 CHAIN "RUN 5/FORGE1"
    180 IF C$="" CHAIN "RUN 5/FORGE2"

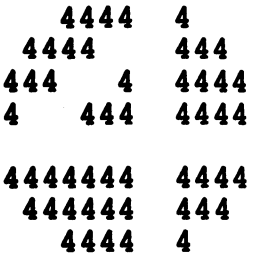
FORGE3
    1310 CHAIN "RUN 5/FORGE1"

FORGE4
    2540 CHAIN "RUN 5/FORGE"
```

FORGE is supplied without charge and without support. Listings can be obtained by dumping the text files listed above.

A FORGE User's guide pamphlet is also enclosed.

5:51:MMH:S105
POINT 4 Data Corporation



T E C H N I C A L M E M O R A N D U M

TO: All IRIS Users

FROM: Applications Software Department

DATE: September 4, 1981

SUBJ: U.CHANGE utility

The following files are the text file versions for the U.CHANGE utility. They should be loaded into BASIC and saved on logical unit 5 as program names represented by the text file name without the prefix "T.".

Text File	Saved As
T.U.CHANGE	5/U.CHANGE
T.U.CHANGE1	5/U.CHANGE1

If an installation requires that the U.CHANGE utility reside on a logical unit other than 5, the following program line numbers should be appropriately modified:

U.CHANGE

```
2260 LET I$=I$,"^ _ [<00> ",E$,"!]\215\RUN 5/U.CHANGE1"
```

U.CHANGE1

```
590 IF I$="Y" CHAIN "RUN 5/U.CHANGE"
2690 LET C$=C$,"\215\\215\\215\\215\\215\RUN 5/U.CHANGE1"
```

The U.CHANGE utility is supplied without charge and without support. Listings can be obtained by dumping the text files listed above.

A U.CHANGE User's guide pamphlet is also enclosed.

MMH/km

9:91:MMH:S109
POINT 4 Data Corporation

T E C H N I C A L M E M O R A N D U M

TO: All IRIS Users

FROM: Applications Software Department

DATE: September 4, 1981

SUBJ: U.PROTECT utility

The following files are the text file versions for the U.PROTECT utility. They should be loaded into BASIC and saved on logical unit 5 as program names represented by the text file name without the prefix "T.".

Text File	Saved As
T.U.PROTECT	5/U.PROTECT
T.U.PROTECT1	5/U.PROTECT1

If an installation requires that the U.PROTECT utility reside on a logical unit other than 5, the following program line numbers should be appropriately modified:

U.PROTECT

```
1720 LET I$=I$,"^ _ [<00> ",E$,"!]\215\RUN 5/U.PROTECT1"
```

U.PROTECT1

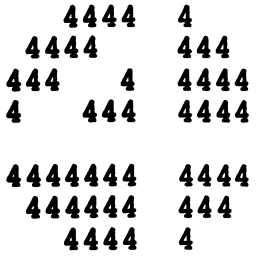
```
580 IF I$="Y" CHAIN "RUN 5/U.PROTECT"
2700 LET C$=C$,I$[6],"!\215\RUN 5/U.PROTECT1"
```

The U.PROTECT utility is supplied without charge and without support. Listings can be obtained by dumping the text files listed above.

A U.PROTECT User's guide pamphlet is also enclosed.

MMH/km

8:81:MMH:S108
POINT 4 Data Corporation



T E C H N I C A L M E M O R A N D U M

TO: All IRIS Users

FROM: Applications Software Department

DATE: September 4, 1981

SUBJ: U.SAVE utility

The following files are the text file versions for the U.SAVE utility. They should be loaded into BASIC and saved on logical unit 5 as program names represented by the text file name without the prefix "T.".

Text File	Saved As
T.U.SAVE	5/U.SAVE
T.U.SAVE1	5/U.SAVE1

If an installation requires that the U.SAVE utility reside on a logical unit other than 5, the following program line numbers should be appropriately modified:

U.SAVE

```
1720 LET I$=I$,"^ _ [<00> ",E$,"!]\215\RUN 5/U.SAVE1"
```

U.SAVE1

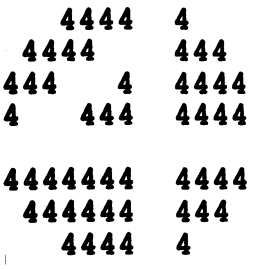
```
580 IF I$="Y" CHAIN "RUN 5/U.SAVE"
2700 LET C$=C$,I$[6],"!\215\RUN 5/U.SAVE1"
```

The U.SAVE utility is supplied without charge and without support. Listings can be obtained by dumping the text files listed above.

A U.SAVE User's guide pamphlet is also enclosed.

MMH/km

7:81:MMH:S107
POINT 4 Data Corporation



T E C H N I C A L M E M O R A N D U M

TO: All IRIS Users

FROM: Applications Software Department

DATE: September 4, 1981

SUBJ: U.COPY utility

The following files are the text file versions for the U.COPY utility. They should be loaded into BASIC and saved on logical unit 5 as program names represented by the text file name without the prefix "T.".

Test File	Saved As
T.U.COPY	5/U.COPY
T.U.COPY1	5/U.COPY1

If an installation requires that the U.COPY utility reside on a logical unit other than 5, the following program line numbers should be appropriately modified:

U.COPY

```
2290 LET I$=I$,"^ _ [<00> ",E$,"!]\215\RUN 5/U.COPY1"
```

U.COPY1

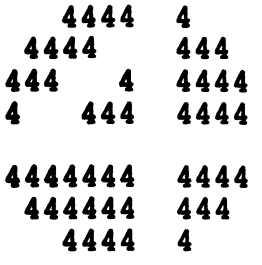
```
600 IF I$="Y" CHAIN "RUN 5/U.COPY"
2700 LET C$="0/COPY ",D2$,"/",I$[4],"!=",N$,"/",I$[4],"\215\RUN
5/U.COPY1"
```

The U.COPY utility is supplied without charge and without support. Listings can be obtained by dumping the text files listed above.

A U.COPY User's guide pamphlet is also enclosed.

MMH/km

6:61:MMH:S106
POINT 4 Data Corporation



T E C H N I C A L M E M O R A N D U M

TO: All IRIS Users

FROM: Applications Software Department

DATE: September 4, 1981

SUBJ: U.KILL utility

The following files are the text file versions for the U.KILL utility. They should be loaded into BASIC and saved on logical unit 5 as program names represented by the text file name without the prefix "T.".

Text File	Saved As
T.U.KILL	5/U.KILL
T.U.KILL1	5/U.KILL1

If an installation requires that the U.KILL utility reside on a logical unit other than 5, the following program line numbers should be appropriately modified:

U.KILL

```
1970 LET I$=I$,"^ _ [<00> ",E$,"!]\215\RUN 5/U.KILL1"
```

U.KILL1

```
580 IF I$="Y" CHAIN "RUN 5/U.KILL"
2680 LET C$="KILL ",N$,"/",I$[4],"\215\RUN 5/U.KILL1"
```

The U.KILL utility is supplied without charge and without support. Listings can be obtained by dumping the text files listed above.

A U.KILL User's guide pamphlet is also enclosed.

MMH/km

12:81:EVW:0363
POINT 4 Data Corporation

```
   4444  4
  4444   444
 444   4 4444
 4   444 4444

 4444444 4444
  444444 444
    4444 4
```

T E C H N I C A L M E M O R A N D U M

TO: All IRIS R7.5 Televideo 950 CRT Users

FROM: Ed Van Wig EVW

DATE: December 14, 1981

SUBJ: Port Type for Televideo 950 Terms driver

A pre-released version of the Televideo 950 terms driver was sent out under Revision 7.4. This version has a port type code of 15. This has been changed for Revision 7.5 to a port type of 4.

technical memo

Purpose

To provide a cursor tracking facility for BASIC data entry programs.

Functional Capability

This cursor tracking facility supplies a method other than field oriented BASIC data entry programs. It also allows the full screen editing capabilities previously available only with the 3270 type protocols.

Implementation

To implement this feature, simply enable the $TERM module with the proper cursor tracking for your CRT.

Octal Control Code

Once implementation is done, any BASIC input statement containing the octal control code "\001\" will then automatically invoke this feature. For example:

```
70 INPUT @15,8;"\001\"I$
```

The user will enter data into "I$", which contains any cursor control characters produced at the keyboard. The application program will interpret the input string and act appropriately. The octal control codes produced in this mode are as follows:

OCTAL CODE	CONTROL KEY
010	Left Arrow
040	Right Arrow
052	Down Arrow
053	Up Arrow

Example

On page three is a sample data entry routine which makes use of cursor tracking. When a program calls this routine, the terminal becomes a "scratch pad" on which the user may write, using the left, right, up and down arrow keys to move the cursor. When the user presses the "RETURN" key, the calling program receives a work variable (S$), which contains the screen contents as follows:

Position	Contents
0001-0080	CRT line one
0081-0160	CRT line two
"	"
"	"
1841-1919	CRT line 24

Note

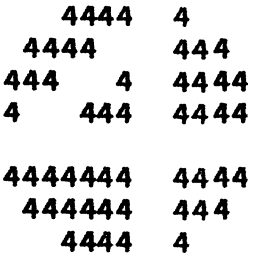
In the following example, the screen work variable is dimensioned one byte short of the 1920 characters which will fit on a 24 X 80 CRT. This is done since entry of data to the last position on the screen might cause the image to "roll up" on some terminals. For a CRT with roll mode disabled, this routine could be modified to allow the use of all 1920 character positions.

5:15:TM:24
Attachment

SAMPLE DATA ENTRY ROUTINE USING CURSOR TRACKING

```
5000 REMACRO; SET, L=5000                                   !*ENTRY POINT
5000 REM
5000 DIM S$[1919], I$[136]
5010 REM S$ - SCREEN BUFFER FOR 24 x 80 CHARACTER CRT OPERATING IN ROLL MODE
5010 REM I$ - INPUT WORK STRING FOR PORT WITH BUFFER SIZE 207 OCTAL IN $MMU.
5010 REM
5010 REM ---> SET UP FOR ROUTINE.
5010 REM
5010 PRINT 'CS'                                             !*CLEAR THE SCREEN
5020 LET S$=" ",S$                                          !*BLANK-FILL SCREEN BUFFER
5030 LET H=0                                                !*INITIAL HORIZONTAL POSITION
5040 LET V=0                                                !*INITIAL VERTICAL POSITION
5050 REM
5050 REM ---> GET DATA USING CURSOR TRACKING.
5050 REM
5050 INPUT @H,V;"\001\"I$                                   !*INPUT AT CURRENT POSITION
5060 REM
5060 REM ---> INTERPRET THE INPUT WORK STRING.
5060 REM
5060 FOR P=1 TO LEN I$                                      !*FOR EACH CHARACTER ...
5070   LET C=H+V                                            !*SAVE SUM OF CO-ORDINATES
5080   IF I$[P,P]="\040\" LET H=H+1                         !*RIGHT ARROW ?
5090   IF I$[P,P]="\010\" LET H=H-1                         !*LEFT ARROW ?
5100   IF I$[P,P]="\052\" LET V=V+1                         !*DOWN ARROW ?
5110   IF I$[P,P]="\053\" LET V=V-1                         !*UP ARROW ?
5120   IF H+V<>C GOTO 5160                                  !*IF CURSOR MOVED, JUMP AHEAD
5130   LET X=(V+1)*80-79+H                                  !*STORE BYTE IN SCREEN BUFFER
5140   REM
5140   IF X<1920 IF I$[P,P]>=" " LET S$[X,X]=I$[P,P]
5150   REM
5150   LET H=H+1                                            !*CURSOR MOVES FORWARD ONE
5160   IF H>=0 GOTO 5200                                    !*CHECK MINIMUM HORIZONTAL
5170   LET H=79                                               !*WRAP-AROUND IF LESS...
5180   LET V=V-1                                              !*AND MOVE UP ONE LINE
5190   GOTO 5230                                              !*THEN SKIP FORWARD
5200   IF H<=79 GOTO 5230                                   !*CHECK MAXIMUM HORIZONTAL
5210   LET H=0                                                !*WRAP-AROUND IF GREATER...
5220   LET V=V+1                                              !*AND MOVE DOWN ONE LINE
5230   IF V<0 LET V=23                                      !*CHECK MINIMUM VERTICAL
5240   IF V>23 GOTO 5290                                    !*CHECK MAXIMUM VERTICAL
5250 NEXT P                                                 !*GET NEXT CHARACTER
5260 REM
5260 REM ---> IF STRING FULL, GET MORE DATA WITH CURSOR TRACKING.
5260 REM      OTHERWISE, "RETURN" WAS PRESSED, SO EXIT ROUTINE.
5260 REM
5260 IF LEN I$-136 GOTO 5050                                !*CHECK FOR "RETURN"
5270 PRINT 'CS';S$;                                         !*RE-DISPLAY SCREEN
5280 RETURN                                                 !*RETURN TO CALLER
5290 REM
5290 REM ---> OVERFLOW:  VERTICAL POSITION EXCEEDED MAXIMUM.
5290 REM
5290 LET V=23                                               !*RESET VERTICAL POSITION
5300 PRINT 'CSRB';S$;                                       !*RE-DISPLAY SCREEN
5310 GOTO 5050                                              !*RETURN TO DATA ENTRY
```

POINT 4 Data Corporation



T E C H N I C A L M E M O R A N D U M

TO: IRIS Users (MARK 5)

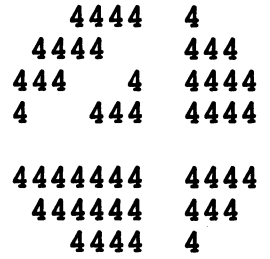
FROM: POINT 4 Customer Support

DATE: December 30, 1981

SUBJ: Blockcopy and CTU Multiblock Set-up

Certain disc controller/disc drive combinations (e.g., MCT TDC-802/Calcomp T80) require changes to the BZUP within BLOCKCOPY or CTU Multiblock before the utility can be used. To determine if such a change is required, refer to the DISC SPECIFICATION sheet in your Peripherals Handbook relating to your configuration. If the section titled "Set-up Parameters" has none listed then you are finished. If there is a list of changes to be made, look at the address indicated within the utility (BLOCKCOPY or MULTIBLOCK) and if the "old contents" listed match the contents in the utility, change to the listed "new contents".

POINT 4 Data Corporation



T E C H N I C A L M E M O R A N D U M

TO: MARK 3 BASF PSS Users

FROM: Customer Support

DATE: January 27, 1982

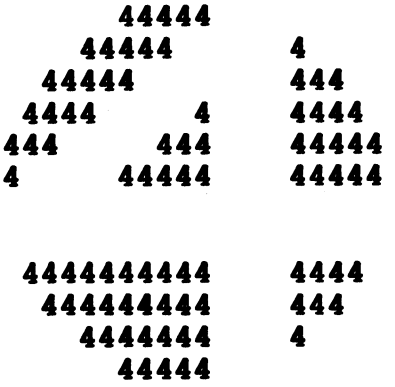
SUBJ: Saving Logical Unit Zero

Using DISCUTLITY 2.5, it is possible to SAVE logical unit zero only. To do this, specify the SAVE program and the BASF disc, then the program will prompt for SURFACE, answer: OPT,0,1,2 followed by a starting cylinder of zero and for number of cylinders specify 40. When restoring an LU 0 tape follow the same procedure (only use RESTORE) and note that the number of cylinders will not be requested (whatever is on the tape will be restored).

We recommend that your LU/0 be saved on a tape in addition to a tape of the entire disc. This will allow more flexibility if you need to restore data.

LMS:kmm

POINT 4 Data Corporation



TECHNICAL MEMORANDUM

TO: ALL IRIS LOTUS CACHE MEMORY (LCM) USERS

FROM: IRIS CUSTOMER SUPPORT

DATE: February 3, 1982

SUBJ: EFFECTS OF LCM SOFTWARE USING SHUTDOWN, REMOVE AND CLEANUP

SHUTDOWN

When running with an LCM activated, SHUTDOWN may take longer to complete than when the LCM is not active. This is due to the time required to write the "updated" blocks stored in the LCM out to disc.

NOTE

DO NOT halt the processor during SHUTDOWN or you may cause loss of data.

REMOVE

REMOVE may take longer to run if blocks from the LU being REMOVE'd are in the LCM.

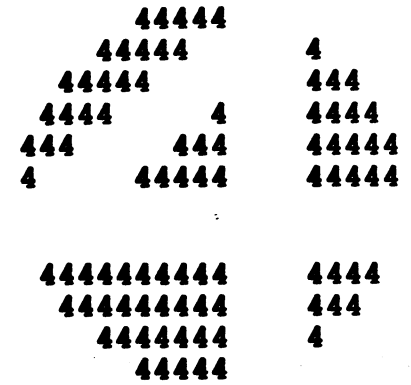
NOTE

DO NOT halt the processor during REMOVE or you may cause loss of data.

CLEANUP

Also note that CLEANUP may not be run when the LCM is active.

POINT 4 Data Corporation



TECHNICAL
MEMORANDUM

TO: ALL IRIS USERS

FROM: IRIS CUSTOMER SUPPORT

DATE: February 3, 1982

SUBJ: $COM - SERIAL COMMUNICATION DRIVER ON R7.5

I. INTRODUCTION

$COM provides a means of transferring text data records over a serial link. (A record is defined as a group of printable and permissible control characters, terminated by a carriage return.)

The driver's primary purpose is to supply a method of interface to Link Management Units (LMUs) or protocol converters. It may also be used in other serial communication applications, such as direct communication between two IRIS systems with POINT 4 MIGHTY MUX multiplexers.

The driver's structure is similar to $LPT, in that it is a noninteractive driver implemented under IRIS and currently works only in conjunction with the POINT 4 MIGHTY MUX or the MARK 3 MUX. It sends and receives text records and uses X-ON/X-OFF as the flow control method (thus no modem control lines are required). There is a read time-out function that allows the user to recover from a stalled condition (driver returns with an Error 49 after three seconds of no data). Other than that, communication is accomplished by simple reads and writes to the driver. It is strictly a half-duplex driver but will respond to the flow control characters (X-ON/X-OFF) at any time.

The driver can be configured for:

7 Data Bits
1 or 2 Stop Bits
Even or Odd Parity
Any MUX-selectable Baud Rate

II. SUPPORTED LMUs

$COM currently supports the following Link Management Units:

LMU	$COM Driver
MICOM - Model 500	COM.A
AIR LAND - Model PCU-200	COM.A
DATALYNX - Model 2780/3780	COM.D*

*COM.D is used for the DATALYNX LMU; all other LMUs will probably use COM.A.

III. $COM CONTROL CODES

The control codes which may be used with $COM are shown in Table 1.

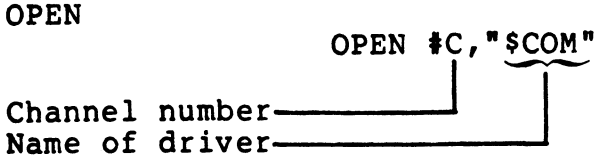
Control codes that are "passed" are not translated to a backslash \257\. Control codes that cannot be passed are translated to a backslash. Take care not to pass a code that a Link Management Unit may require for control purposes. Reserved control codes must remain unchanged.

TABLE 1. PERMISSIBLE CONTROL CODES
(Example for COM.A Version)

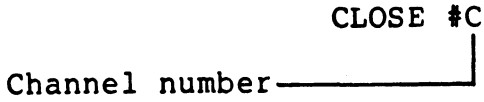
Code	Passed	Translated to a Backslash \257\	Reserved (Used by Driver)
00-NULL		X	
01-SOH		X	
02-STX		X	
03-ETX			X
04-EOT		X	
05-ENQ		X	
06-ACK		X	
07-BEL			X
10-BS	X		
11-HT	X		
12-LF	X		
13-VT	X		
14-FF	X		
15-CR	X		
16-SO	X		
17-SI	X		
20-DLE		X	
21-XON			X
22-DC2		X	
23-XOFF			X
24-DCH		X	
25-NAK		X	
26-SYN		X	
27-ETB		X	
30-CAN		X	
31-EM		X	
32-SUB	X		
33-ESC	X		
34-FS	X		
35-GS		X	
36-RS		X	
37-US		X	

IV. DRIVER I/O STATEMENT FORMATS

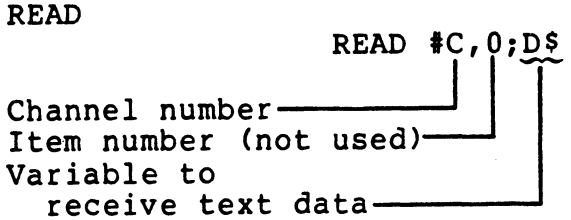
A. OPEN



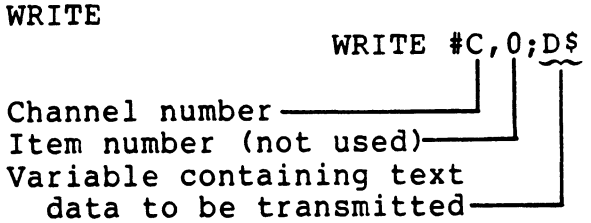
B. CLOSE



C. READ



D. WRITE



V. POSSIBLE ERROR CONDITIONS

A. OPEN

Error 72 - Unable to open file; or port is configured incorrectly

B. CLOSE

No possible error conditions

C. READ

Error 40 - Received record is too large
Error 49 - Read has timed-out before record was received
Error 65 - Item in read statement is not a string variable
Error 70 - Mux overload

D. WRITE

Error 65 - Item in write statement is not a string variable

NOTE

A power-fail Error 74 can be received at any time.

If a power-fail occurs, data may be lost; your application should provide means for checking and recovery.

VI. LOCAL LINK

Local Link requires POINT 4 MIGHTY MUX multiplexers or the MARK 3 MUX.

A. POINT 4-TO-POINT 4 CABLING (see Figure 1, Cable A)

$COM ports should be connected by a cable as described below:

1. MARK 3-to-MARK 3 Cable - 6-pin Molex
   - Pin 6 to Pin 6
   - Pin 1 to Pin 4
   - Pin 4 to Pin 1

2. MARK 5 or MARK 8-to-MARK 5 or MARK 8 Cable - 25-pin Connector
   - Pin 7 to Pin 7
   - Pin 2 to Pin 3
   - Pin 3 to Pin 2

3. MARK 3-to-MARK 5 or MARK 8 Cable - 6-pin Molex-to-25-pin Connector
   - Pin 6 to Pin 7
   - Pin 4 to Pin 3
   - Pin 1 to Pin 2

B. LOCAL DATA TRANSFER PROCEDURE

1. Insure correct configuration (see Figure 1).

2. Execute Receive Data utility (see Receive program example in Appendix A).

3. Execute Transmit Data utility (see Transmit program example in Appendix B).

4. Wait for end of data (indicated by a \207\).

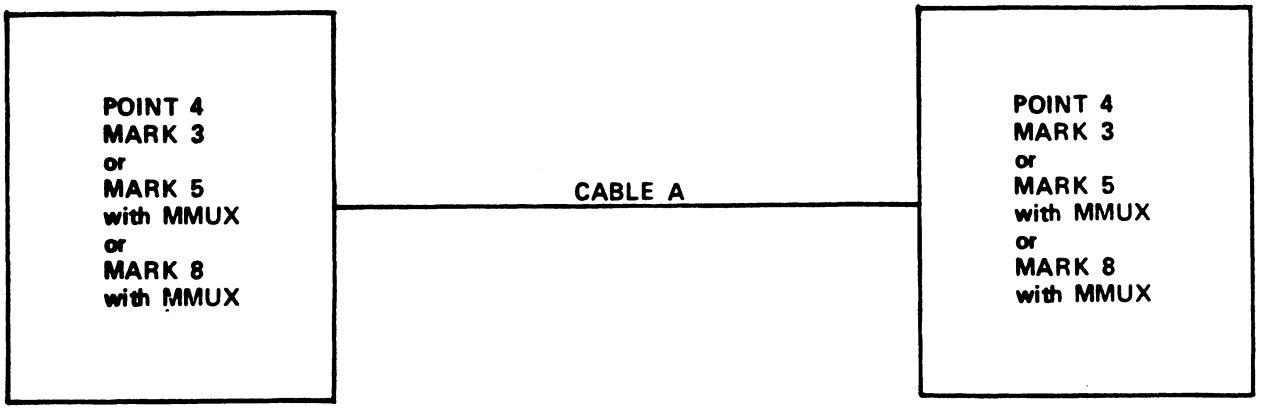


Figure 1. Local Link for $COM

VII. REMOTE LINK

A. POINT 4-TO-LMU CABLING

Same as POINT 4-to-POINT 4 Cabling
(see Figure 2, Cable A)

NOTE

Some Link Management Units require DTR to be on. If so, connect pin 5 (on Mux side) to pin 20 (on LMU side). Keep in mind that you must also set the PCON word (Bit 3 to a 1).

B. LMU-TO-MODEM CABLING

Standard RS232-C 25-pin connector - all pins connected
(see Figure 2, Cable B)

C. REMOTE DATA TRANSFER PROCEDURE

1. Insure correct configuration (Figure 2).
2. Reset LMUs.
3. Execute Receive Data utility (see Receive program example in Appendix A).
4. Dial up to receiving computer and establish the connection.
5. Execute transmit data utility (see Transmit program example in Appendix B).
6. Wait for end of data (indicated by a \207\). If no more data is to be sent, then reset the LMUs. This should cause the phones to hang-up.

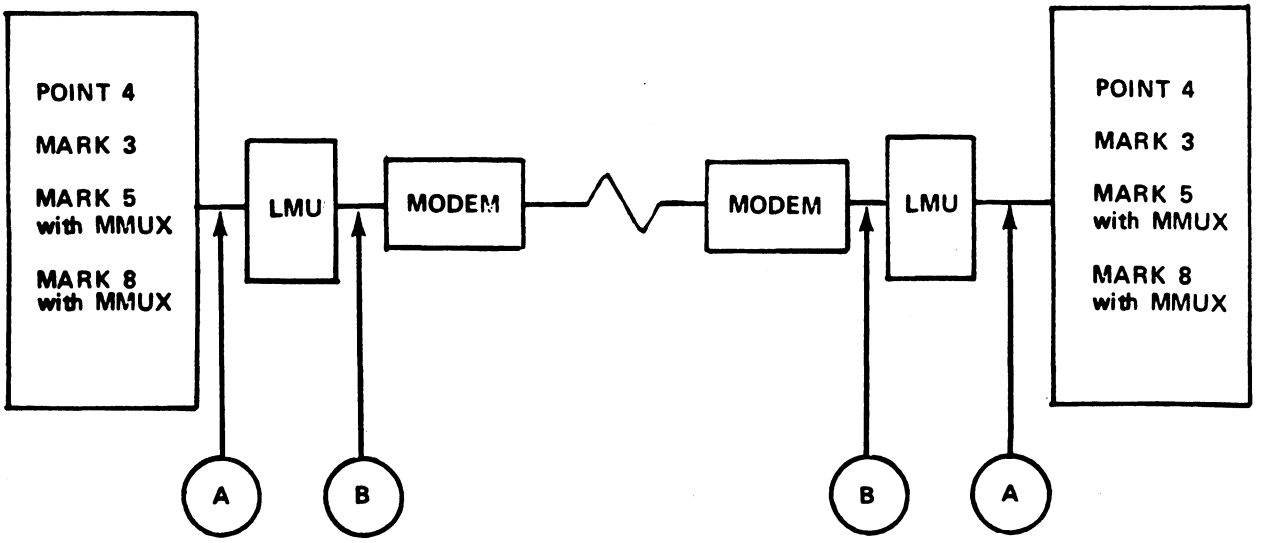


Figure 2. Remote Link for $COM

VIII. SYSTEM PREPARATION

A. MARK 5 and MARK 8 Procedure

1. Enable appropriate COM driver; e.g., CHANGE COM.A to $COM.

2. Patch $COM. If a port other than port 3 is to be used as the communications port, then patch $COM as follows:

Location	Patch	Old Contents
10271	7	3

In this example, physical Mux port 7 is the port to be used by $COM.

3. Patch the ATRIB table in $MUX so that the entry corresponding to the $COM port contains:

```
0       1         ; 1 port of this type
1       40057*    ; PCON parameters
2       130       ; largest record plus 10(8) bytes
3       0
4       0
5       0
6       0
7       0
```

*In this example, a PCON of 40057 means:
7 bits
1 stop bit
9600 Baud
Even parity

The PCON word for your device may be different (see IRIS Manager Manual, p. 11-4).

4. Re-IPL to effect changes.

B. MARK 3 Procedure

1. Enable appropriate COM driver, e.g., CHANGE COM.A to $COM.

2. Use Port 3 as the communications port.

3. Patch the ATRIB table in $MUX so that the entry corresponding to the $COM port #3 contains:

```
0       1           ; 1 port of this type
1       120211*     ; PCON parameters
2       130         ; largest record plus 10(8) bytes
3       0
4       0
5       0
6       0
7       0
```

*In this example, a PCON of 120211 means:
7 bits
1 stop bit
Even parity

NOTE

The Baud rate is determined by a jumper on the PIB board.

The PCON word for your device may be different (see Port Control Word for MARK 3, Section 33).

4. Re-IPL to effect changes.

APPENDIX A

EXAMPLE OF TRANSMIT DATA UTILITY

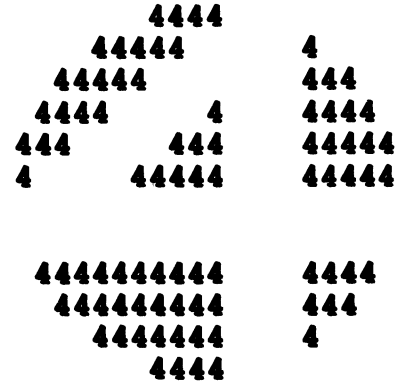
```
1 REM R75SEND.COM
10 REM
20 REM-PROGRAM TO TRANSFER A FILE USING $COM
30 REM
40 REM
50 REM-FILENAME LENGTH
60 REM
70 DIM F$[14]
80 REM
90 REM-MAXIMUM RECORD LENGTH
100 REM
110 DIM A$[80]
120 IF  ERR 0 GOTO  460
130 OPEN #0,"$COM"
140 IF  ERR 0
150 REM
160 REM-FILES WITH UNACCEPABLE CONTROL CODES WILL
170 REM-BE TRANSLATED TO A SLASH (/)
180 REM-GENERALLY, FORM CONTROL CODES
190 REM-WILL BE PASSED.
200 REM
210 INPUT "ENTER FILENAME TO BE TRANSMITTED:  "F$
220 PRINT "\215\"
230 IF  ERR 0 GOTO  490
240 OPEN #1,F$
250 IF  ERR 0
260 REM
270 REM-READ RECORD FROM FILE, AND DISPLAY IT
280 REM
290 READ #1;A$
300 PRINT A$;
310 IF A$="" GOTO 550
320 REM
330 REM-WRITE DATA USING $COM
340 REM-PROGRAM EXECUTION WILL STOP (RECORD-LOCKED) UNTIL
350 REM-DATA ARE WRITTEN. A RECORD IS FORMED BY A CARRIAGE RETURN.
360 REM-ANY ERROR CONDITION SHOULD BE VIEWED AS A TERMINAL
370 REM-CONDITION, AND A RETRANSMISSION SHOULD BE PERFORMED
380 REM
390 IF  ERR 0 GOTO  620
400 WRITE #0,0;A$
410 IF  ERR 0
420 GOTO  290
430 REM
440 REM
450 REM
460 PRINT "UNABLE TO OPEN $COM"
470 CHAIN ""
480 REM
490 PRINT "\215\UNABLE TO OPEN  "F$
500 CLOSE #0
510 CHAIN ""
520 REM
530 REM-END OF DATA CODE
540 REM
550 WRITE #0,0;"\203\"
560 PRINT "FILE TRANSFERRED"
570 SIGNAL 3,0
580 CLOSE #1
590 CLOSE #0
600 CHAIN ""
610 REM
620 PRINT SPC (8),"ERROR"
630 CLOSE #0
640 CLOSE #1
650 CHAIN ""
```

APPENDIX B

EXAMPLE OF RECEIVE DATA UTILITY

```
1 REM R75TRANS.COM
10 REM
20 REM-PROGRAM TO RECEIVE A FILE AND SAVE IT ON DISC USING $COM
30 REM
40 DIM C
50 REM-FILENAME LENGTH
60 REM
70 DIM F$[14]
80 REM
90 REM-MAXIMUM RECORD LENGTH
100 REM
110 DIM A$[80]
120 IF  ERR 0 GOTO  600
130 OPEN #0,"$COM"
140 IF  ERR 0
150 REM
160 REM-FILE TO BE CREATED
170 REM
180 INPUT "ENTER NAME OF FILE TO BE CREATED:    "F$
190 PRINT "\215\"
200 IF  ERR 0 GOTO  630
210 BUILD #1,+F$
220 IF  ERR 0
230 LET C=0
240 REM
250 REM-READ RECORD FROM $COM AND WRITE IT IN FILE
260 REM-PROGRAM EXECUTION WILL BE STOPPED (RECORD LOCKED)
270 REM-UNTIL DATA ARE RECEIVED, OR A RECEIVER TIME OUT
280 REM-OCCURS(10 SECOND TIME OUT).
290 REM-A RECORD IS FORMED BY A CARRIAGE RETURN,
300 REM-OR BUFFER FULL (DIM OF STRING)
310 REM-ANY ERROR CONDITION SHOULD BE VIEWED AS A TERMINAL
320 REM-CONDITION AND A RETRANSMISSION SHOULD BE PERFORMED.
330 REM-PARITY ERRORS WILL BE CONVERTED TO A "\" CHARACTER.
340 IF  ERR 0 GOTO  670
350 READ #0,0;A$
360 IF  ERR 0
370 LET C=0
380 REM
390 REM-ANY CONTROL CODES RECEIVED AFTER A CARRIAGE RETURN
400 REM-WILL BE ADDED TO THE BEGINNING OF THE NEXT RECORD.
410 REM-IF USING A PROTOCAL CONVERTER EXTRA LINE FEED CODES
420 REM-MAY BE ADDED TO THE TEXT AND SHOULD BE DELETED.
430 REM
440 IF A$[1,1]="\212\" LET A$=A$[2, LEN A$]
450 REM
460 REM-END OF DATA CODE
470 REM
480 IF A$[1,1]="\207\" GOTO 840
490 IF  ERR 0 GOTO  900
500 WRITE #1;A$
510 IF  ERR 0
520 REM
530 REM-DISPLAY IT ON CONSOLE
540 REM
550 PRINT A$;
560 GOTO  340
570 REM
580 REM
590 REM
600 PRINT "UNABLE TO OPEN $COM"
610 CHAIN ""
620 REM
630 PRINT "\215\UNABLE TO OPEN:    "F$
640 CLOSE #0
650 CHAIN ""
660 REM
670 IF  SPC (8)=40 GOTO 740
680 IF  SPC (8)=49 GOTO 770
690 PRINT  SPC (8),"ERROR"
700 CLOSE #0
710 CLOSE #1
720 CHAIN ""
730 REM
740 PRINT "BUFFER OVERFLOW"
750 GOTO  700
760 REM
770 LET C=C+1
780 IF C=1 PRINT "\215\"
790 PRINT "\213\READER TIME OUT",C
800 IF  ERR 0
810 SIGNAL 3,0
820 GOTO  340
830 REM
840 PRINT "END OF FILE"
850 SIGNAL 3,0
860 CLOSE #0
870 CLOSE #1
880 CHAIN ""
890 REM
900 PRINT "DISC WRITE ERROR"
910 CLOSE #0
920 CLOSE #1
930 CHAIN ""
```

POINT 4 Data Corporation



TECHNICAL MEMORANDUM

TO: All IRIS 7.5 MARK 3 Users

FROM: IRIS Customer Support

DATE: May 10, 1982

SUBJ: INSTALLING MORE THAN ONE LINE PRINTER ON A SYSTEM OR CHANGING FROM CURRENT PORT TO ANOTHER.

The MARK 3 LPTM driver has been assembled to communicate with port three only. To run a Line Printer on another port or to have more than one printer on a system, changes must be made to the LPTM driver. These changes must be made using DSP.

To install a second Line Printer or to change LPT to a new port, the locations to be changed are as follows:

1. First, change LPTM's TYPE (word 10 in the header block) from 36 to 3.

2. Make another copy of LPTM.

3. On the copy of LPTM, change the TYPE back to 36.

4. Make the following changes:

LOC	New Contents	Old Contents
10206	0,1,2,3,4,5 or 6	3
10232	610XX	61016
10254	610XX	61016

where XX denotes one of the following device codes:

10 - Port Zero	20 - Port Four
12 - Port One	22 - Port five
14 - Port Two	24 - Port Six
16 - Port Three	

5. Run GUIDE to get the PCB address.

POINT 4
DATA CORPORATION

SOFTWARE CHANGE ORDER

software production manager ONLY

STATUS _____

PROG SYSTEM
UPDATE BY _____ DATE _____

SCO # _____ DATE _____

MASTER FILE
UPDATE BY _____ DATE _____

MASTER
FILENAME _____

PATCH # DSB375-02

PRODUCT DISCSUBS #3

DETAIL SEARCH

ASM DATE 06-13-77 RELEASE # 7.5

UPDATE DATE 03-08-82 BY DSZ

REVIEW DATE _____ BY _____

PR # _____ PROBLEM SEARCH mode 2 won't put terminator byte in string if full key length used.

SPECIAL INSTRUCTIONS (other products?) OPTIONAL PATCH - DISCUSUB # 61 must be core resident if this patch is entered.

LOCATION (OCTAL)	NEW CONTENTS (OCTAL AND/OR SYMBOLIC)	COMMENTS (DESCRIBE SOLUTION)	OLD CONTENTS
27100	LDA 1,63	; Mask for Zeroing byte	14173
27101	JMP 26772	; to patch	773
26772	DSZ 173	; End of Key?	Unused
26773	JMP 27074	; No	
26774	LDA 0,0,3	; Yes grab next word	
26775	AND 1,0	; Zero left byte	
26776	STA 0,0,3	; store in partition	
26777	JMP 27102	; back from patch	Unused

Form 2031

POINT 4 DATA CORPORATION

SOFTWARE CHANGE ORDER

software production manager ONLY

STATUS _____

PROD. SYSTEM UPDATE BY _____ DATE _____

SCO # _____ DATE _____

MASTER FILE UPDATE BY _____ DATE _____

MASTER FILENAME _____

PATCH # MK3SOV-01

PRODUCT ALL MARK 3 SOV'S (IN REX)

DETAIL _____

ASM DATE _____ RELEASE # _____

UPDATE DATE 04-12-82 BY JAS

REVIEW DATE 04-12-82 BY LMS

PR # _____ PROBLEM Trap 3 with (A2)=1. Recal. routine didn't clear previous drive table.

SPECIAL INSTRUCTIONS (other products?) FOR MARK 3 ONLY

LOCATION (OCTAL)	NEW CONTENTS (OCTAL AND/OR SYMBOLIC)	COMMENTS (DESCRIBE SOLUTION)	OLD CONTENTS
667	7513		7532
7514	0		UNUSED
5	STA 3, .-1	Save Return Address	
6	SUBZL 0,0	Drive # Mask =1	
7	LDA 3, 7531	Preset Ptr. to Dr.0 Tab	
7520	LDA 1,1,2	Get Part	
1	AND 0,1, SZR	Drive 0 ?	
2	LDA 3,7532	No, Set Ptr. to D1 Tab	
3	SUB 0,0		
4	STA 0,0,3	Clear Previous Cyl.	
5	STA 0,1,3	Clear Previous Head	
6	LDA 3,7514	Restore A3.	
7	JMP @.+1	Recalibrate	
7530	7777		

Form 2031

POINT 4 DATA CORPORATION

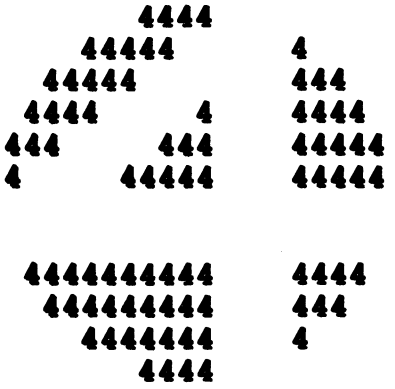
SOFTWARE CHANGE ORDER

PATCH # MK3SOV-01

PRODUCT ALL MARK 3 SOV'S (IN REX) ASM DATE _____ RELEASE # _____

LOCATION (OCTAL)	NEW CONTENTS (OCTAL AND/OR SYMBOLIC)	COMMENTS (DESCRIBE SOLUTION)	OLD CONTENTS
7531	10173	Drive 0 Table Ptr.	UNUSED
2	10176	Drive 1 Table Ptr.	
7564	JMP 7515	To Patch	577

POINT 4 Data Corporation



TECHNICAL MEMORANDUM

TO: All IRIS Users

FROM: IRIS Customer Support

DATE: May 12, 1982

SUBJ: SKIP COMMANDS IN DSP

In using DSP there is some confusion in "checking contents" at a location where you have just "entered new contents." This confusion usually involves the "skip" instructions as listed below. For example if you enter

```
SEQ 0,3
```

DSP will display something else. When you check the contents entered above, DSP will display

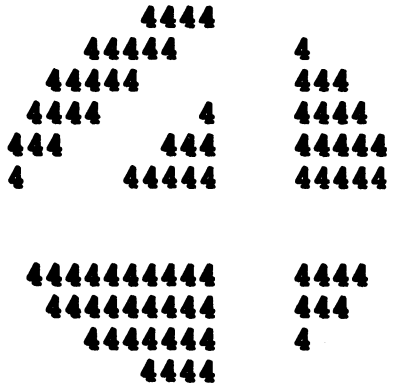
```
SUB# 0,3,SZR
```

These are, however, equivalent forms.

WHAT YOU ENTER	WHAT DSP WILL DISPLAY
SEQ X,Y	SUB# X,Y,SZR
SNE X,Y	SUB# X,Y,SNR
SGR X,Y	SUBZ# X,Y,SZC
SLS X,Y	ADCZ# X,Y,SNC
SGE X,Y	ADCZ# X,Y,SZC
SLE X,Y	SUBZ# X,Y,SNC
SKZ X,X	MOV# X,X,SZR
SNZ X,X	MOV# X,X,SNR
SSP X,X	MOVL# X,X,SZC
SSN X,X	MOVL# X,X,SNC
SKO X,X	MOVR# X,X,SNC
SKE X,X	MOVR# X,X,SZC

where X and Y are digits 0 through 3

POINT 4 Data Corporation



T E C H N I C A L
M E M O R A N D U M

TO: All IRIS 7.5 MARK 3 Users

FROM: IRIS Customer Support

DATE: May 13, 1982

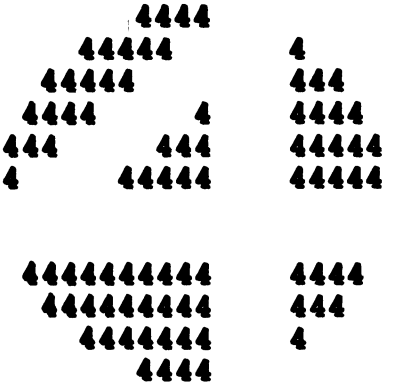
SUBJ: PORT CONTROL WORD (PCW) FOR MARK 3

The Port Control Word for the MUX on the MARK 3 is constructed differently from that on the MARK 5. Table 33-1 provides PCW values for the various combinations of number of data bits, parity, and number of stop bits. Select the appropriate PCW value for your system.

TABLE 33-1. PCW VALUES

No. of Data Bits	Parity	No. of Stop Bits	PCW Value
7	Even	2	140201
7	Odd	2	140205
7	Even	1	140211
7	Odd	1	140215
8	Inhibited	2	140221
8	Inhibited	1	140225
8	Even	1	140231
8	Odd	1	140235

POINT 4 Data Corporation



TECHNICAL MEMORANDUM

TO: ALL IRIS USERS

FROM: IRIS Customer Support

DATE: May 13, 1982

SUBJ: Setting Up Line Printers

Selection of the correct driver for your line printer is not based on the particular type of printer you may have on your system. The selection of the driver is based on the interface that connects the line printer to the computer. Possible interfaces are:

- POINT 4 310 MIGHTY MUX
- Data General 4060-type multiplexer
- Device Code 17 Controller Board
- Device Code 51 Controller Board

IRIS supports several types of universal line printer drivers. They are 'universal' because they can be customized to support almost any particular make or model of line printer.

POINT 4 supplies you with all the line printer driver files supported under IRIS. They are read onto your system disc at Sysgen time. First, select the appropriate driver for your system; it must be copied so that there is at least one valid copy of the driver available at any time. The names for these drivers and their specifications are given in Table 34-1.

As an aid to configuring most line printers, POINT 4 supplies a GUIDE module, GUIDE.LPT. It can be accessed from the GUIDE Menu or from the system prompt at any time without taking any special precautions since it makes no changes to any file by itself. It gives information on how to use DSP to set up your line printer.

POINT 4 recommends that the system be backed up before using DSP.

Setting up your line printer requires five steps:

1. Select the appropriate driver from Table 34-1

2. Copy the driver

3. Run GUIDE.LPT

4. Use DSP to apply GUIDE.LPT's output

5. Test and customize your line printer

SELECT APPROPRIATE DRIVER

Since IRIS offers four different drivers, make your choice based on the information given in Table 34-1.

TABLE 34-1. IRIS DRIVERS AND SPECIFICATIONS

Driver Name	Specifications
LPTM	Line printers using the POINT 4 310 MIGHTY MUX. If your line printer has an RS-232 serial option, it may be plugged into a port on the POINT 4 310 Multiplexer. The POINT 4 Mux outputs to the printer using Direct Memory Access (DMA). With the POINT 4 Mux, the CPU does not have to handle individual characters which results in better performance for all users while print jobs are running.
LPTP	Line printers using a device code 17 controller. Device code 17 controllers use PIO and require a special controller board in your computer. The CPU must handle each character using such PIO instructions as DOA or SKPBZ resulting in a slower rate of data transfer than with DMA. Usually, the data is transmitted to the printer in 7 or 8 parallel lines.
LPTP51	Line printers using a device code 51 controller. Same as LPTP.
LPTD	Line printers using a Data General 4060-type multiplexer. Requires the RS-232 option but does not have the DMA advantage.
	NOTE POINT 4's GUIDE.LPT does not give instructions for the installation of an LPTD driver. A listing of the LPTD driver file is included in Appendix A (page 34-10) to assist you in making the necessary modifications.

COPY THE LINE PRINTER DRIVER

It is necessary to make a copy of the driver file to ensure that a valid (unmodified) file of that driver remains on the system.

A legal name for a copy of an IRIS line printer driver may begin with "LPT" and may be followed by a digit. The name may not include periods, other letters, or symbols. See Figure 34-1 for examples of legal and illegal copied line printer driver names.

The procedure for copying a line printer driver is given in Figure 34-2. In that example, the driver to be copied is LPTM and user input is underlined.

Legal Name	Illegal Name
LPT	LPTA
LPT1	LPTP
LPT2	LPTP51
.	LPT.1
.	
.	LPT/3
LPT99	

Figure 34-1. Legal Names for Line Printer Driver Copies

Terminal	Description
#DSP <CTRL-E>key<CTRL-E> F LPTM H	where key is the password
D10	dump the contents of word 10
10: 36 xxx xxx <ESC>	press <ESC> to stop the printout. If the contents of word 10 is not 36, something is wrong. Make sure you started with a valid copy of IRIS.
10: 3	enter 3 into word 10
D10	check that the contents of word 10 is now 3
<CTRL-C>	exit from DSP
#COPY LPT=LPTM	if the name LPT is already in use, you may use the name LPT1, LPT2, LPT3, etc.
COPIED!	
#DSP F LPTM H	
D10	if the contents of word 10 is not 3, something is wrong. Go back to your backup and start over!
10: 3 xxx xxx <ESC>	
D10	if the contents of word 10 is not 36, something is wrong. Go back to your backup and start over!
10: 36	enter 36 into word 10
F LPT	use the name of the file you created with the COPY command
H D10	if the contents of word 10 is not 77003, something is wrong. Go back to your backup and start over!
10: 77003 xxx xxx <ESC>	
10: 36	enter 36 into word 10
D10	if the contents of word 10 is not 36, something is wrong. Go back to your backup and start over!
X	exit from DSP

Figure 34-2. Procedure For Copying A Line-Printer Driver

GUIDE.LPT

GUIDE.LPT is an interactive BASIC program provided by POINT 4 to assist you in setting up your line printer. The program asks questions about your line printer and then tells you how to make the appropriate changes using DSP.

Using GUIDE.LPT

GUIDE.LPT does not make any changes to the driver file. It gives you the information that has to be entered into the driver file. It will not interfere with any system processes and can be run at any time. If you make any mistakes in entering your answers to GUIDE.LPT's questions, press <ESC> and restart the program.

Notes on GUIDE.LPT Questions

The following notes will help you answer the questions asked by GUIDE.LPT:

a. If your chosen driver is LPTP51, answer all the questions as if you were using LPTP (device code 17).

b. For LPTP questions on 'DIA' and 'interrupt after any busy', answer NO if you are not sure of what is required. If you answer YES and you are wrong, the printer may hang up while printing.

c. If, according to specifications, your printer requires a motor-on character in the OPEN list, enter that as the first character in the OPEN list. The motor-off character should be the last character in the CLOSE list.

d. For an automatic formfeed on OPEN and CLOSE, enter 14 in both lists. (For a word processing printer, consult the appropriate installation document.)

e. The following lists are recommended for a system line printer:

```
CR LIST:                    15
                             0
                            12
                             0
                            -1

MULTIPLE CR LIST:            0
                            12
                             0
                            -1

DELAY AFTER SPECIAL CHARS:   0
                             0
                             0
                            -1
```

f. The following lists are recommended for a word processing printer:

```
CR LIST:                    15
                            -1

MULTIPLE CR LIST:           15
                            -1

DELAY AFTER SPECIAL CHARS:  -1
```

g. Some line printers slash zeros and others slash the letter O. GUIDE.LPT asks:

    DO YOU WISH TO PRINT ZERO IN PLACE OF OH AND VICE VERSA?

    Answer YES or NO depending on your line printer and/or requirements.

h. When GUIDE.LPT asks:

    OUTPUT WHERE?

    Press <RETURN> to display the output on your screen. Write down the output and use the information to modify your driver file.

i. When $LPT is working, run GUIDE.LPT again. Press L when the question 'OUTPUT WHERE?' comes up to get a printout of your line printer configuration.

USE DSP TO APPLY GUIDE.LPT'S OUTPUT

Before using DSP, there are two important things to remember:

1. Always backup your system.

2. Never modify your original line printer drivers. Work with a copy of the driver (see Section 4.8.6.2).

GUIDE.LPT asks you to check certain values in your driver file to make sure that you are working with the correct version.

Changes must be made to the Port Definition Table in $MMUX or $DGMX if you are using the LPTM or LPTD drivers. The Port Definition Table in the appropriate Mux driver must have a zero active file size.

Since line printers have a circular buffer, it will reduce overhead if you set up a large I/O buffer size for this port (approximately 500 characters at 9600 baud).

The proper PCW word in the Port Definition Table must be used both for $LPTM and also for your line printer port in the MUX driver.

TEST AND CUSTOMIZE YOUR LINE PRINTER

To test your current setup, first do a SHUTDOWN and IPL, then run the following BASIC program:

```
10 OPEN #0;"$LPT"
20 PRINT #0;"ABCD"
30 PRINT "*";
40 GOTO 20
```

If you get errors for the OPEN statement, it usually means that you made a mistake in the set up. A common mistake is to give the wrong port number to GUIDE.LPT.

"Logical, IRIS System Port#" refers to the decimal number displayed on your screen when you log-on. This number is always different from the "Octal, origin zero" port number for the same physical unit. Port numbers in octal start at zero:

0 - The first possible MUX port

7 - The last port on an 8-port Mux

10 - The first port on the MUX extender

Remember that the active file size for this port must be zero in the MUX driver.

If you do not get any errors after running the BASIC program but the output is wrong or nothing prints, then check the following:

1. If using $LPTM, check the PCW word for the line printer port in $MMUX

   a. If something prints on the line printer every time you select the printer ON or OFF, the PCW bit 11 (ready status) may be set incorrectly.

   b. If garbage prints on the line printer, then the PCW word may have the wrong number of data bits, stop bits, or the wrong parity.

   c. If the asterisks (*) stop printing on the screen, the PCW word may be wrong.

   d. If the asterisks continue to print, then IRIS thinks it is outputting to the line printer instead of the terminal. Recheck the hardware.

2. Check the cables and check that the proper port is plugged in. Some printers require special jumpers at the printer end before the printer will accept data.

Figure 34-3 illustrates the type of line printer problems that can be solved by using the lists in GUIDE.LPT. Explanations for these problems are as follows:

Error 1 - An error at the top of the page may mean:

   a. Not enough delay characters in the DELAY AFTER SPECIAL CHARS list

   b. If this is the first page to be printed, not enough delay in the OPEN list

Error 2,3 - An error at the beginning or at the end of a line often indicates that there are not enough delay characters in the CR list

Error 4 - Notice that one blank line is missing. This is often caused by not enough delay characters either at the start or at the end of the MULTIPLE CR list

Error 5 - If the whole printout is double spaced, it may be caused by having a 12 (linefeed) in the CR list and having the automatic linefeed option set in the hardware. If that is the case, run GUIDE.LPT again to remove the 12 from the CR list

```
BCDEFGHIJKLMNOPQRSTUVWXYZ0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789
ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789  (1)
ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789
ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789
ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789
ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789


ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789  (2)
ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789
ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789  A
BCDEFGHIJKLMNOPQRSTUVWXYZ0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789
ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789
ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789


ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789
ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789
ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789
ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789
ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789
ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789


ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789
ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789
(3) KBCDEFGHIJKLMNOPQRSTUVWXYZ0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789
ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789
ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789
ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789

                                   (4)  (NOTE: one blank line is missing here)
ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789
ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789
ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789
ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789
ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789
ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789


ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789

ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789

ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789  (5)

ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789

ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789

ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789
```

Figure 34-3. Printer Output

Appendix A

DRIVER FILE LISTING FOR A DG 4060-TYPE MUX

```
0/ASM W75LPTDGMX.510,@$LPT1,-X75DEFSPZ,W75LPTDGMXSA
MAY 10, 1982  11:30:46

; "$LPT" == LINE PRINTER DRIVER FOR "IRIS" R7.2, R7.3, R7.4 OR R7.5
; FOR LPT ON DATA GENERAL 4060 MUX PORT
; DATE LAST EDITED  5-10-82

;                All Rights Reserved
;     Copyright (C) 1974, Educational Data Systems
;     Copyright (C) 1980, Educational Data Systems
;     Copyright (C) 1981, Point 4 Data Corporation
;     Copyright (C) 1982, Point 4 Data Corporation
;   This document contains secret and confidential information
;   of Point 4 Data Corporation, and may not be reproduced, used,
;   or disclosed without the prior written permission of Point 4
;   Data Corporation.

              7       SFTYM=7;LPT BUFFER SAFETY MARGIN
              1       .TXTM  1
          10200       .LOC   BPS
;
;
   10200 177777       -1     ;NO INTH
   10201  10622 .ATRB:ATRIB
   10202  10235       FINIS
   10203  10351       WRITE
   10204 177777       -1     ;NO READ WRITE
;
;********** INIT ROUTINE **********
;
   10205  54444 INIT: STA    3,INPFL
   10206  32773       LDA    2,@.ATRB
   10207 151014       SKZ    2,2       ;FIRST INIT AFTER IPL?
   10210   1401       JMP    1,3       ;NO, JUST RETURN
;
;
   10211  34770 INIT1:LDA    3,.ATRB
   10212  21777       LDA    0,PTOFF,3
   10213   6100       CALL
   10214 100023       CPNPP             ;CHANGE PORT # TO PCB PNTR
   10215   6126       TRAPFAULT         ;ILLEGAL PORT #?
   10216 111000       MOV    0,2        ;AC2 NOW = PRINTER PORT PNTR
   10217  21027       LDA    0,AHA.,2   ;SIZE OF ACTIVE FILE
   10220 101014       SKZ    0,0        ;PORT INTERACTIVE ?
   10221    411       JMP    INERR      ;  YES, SHOULDN'T BE
   10222  52757       STA    2,@.ATRB   ;SAVE PCB = 1ST INIT DONE FLAG
   10223  34756       LDA    3,.ATRB
   10224  54523       STA    3,AATRB    ;SECONDARY PNTR TO ATRIB

;INIT COMPLETE: SET OUTPUT CHAR HANDLER ADDRESS FOR MUX

   10225   4425       JSR    INTH-1     ;GET ADDRESS, SKIP RETURN
   10226  20000 C20K: 20000
   10227  55021       STA    3,TON.,2   ;PUT IT IN PCB
   10230  34421       LDA    3,INPFL
   10231   1401       JMP    1,3        ;NORMAL RETURN

   10232  34402 INERR:LDA    3,ERR43
   10233   2416       JMP    @INPFL

   10234     43 ERR43:43                ;INCORRECT ATTRIBUTES
```

```
;FINISHED (CLOSE) ROUTINE

   10235  54513 FINIS:STA    3,RTNAD
   10236  20413        LDA    0,INPFL
   10237 101014        SKZ    0,0       ;IS INIT STILL PENDING
   10240    403        JMP    FINI2     ;YES
   10241  20407        LDA    0,FINCL   ;NO
   10242   4447        JSR    QSTRI     ;SEND THEM
   10243   4503 FINI2:JSR    JPROD     ;"KICK" PRINTER JUST IN CASE
   10244 102400        SUB    0,0
   10245  40404        STA    0,INPFL   ;CLEAR INIT PENDING IF ON
   10246  40440        STA    0,WRICC   ;IN CASE OF USER ESC
   10247   2501        JMP    @RTNAD
;
   10250     23 FINCL:ATRIB-FINIZ
   10251      0 INPFL:0
;
;********** OUTPUT CHARACTERS ROUTINE **********
;
;ON ENTRY AC2=PCB
;
   10252   5401        JSR    1,3       ;LET MUX DO PFRST
   10253   1400 INTH: JMP    0,3       ;NO INPUT ROUTINE
   10254  54426        STA    3,INTHR   ;START OF OUTPUT CHARACTER ROUTINE
   10255  21001        LDA    0,OCW.,2
   10256 101120        MOVZL  0,0
   10257 101220        MOVZR  0,0
   10260  41001        STA    0,OCW.,2  ;CLEAR MUX BUSY FLAG
   10261  25005        LDA    1,OBP.,2  ;OUTPUT BYTE POINTER
   10262  21004        LDA    0,IBP.,2  ;INPUT BYTE POINTER
   10263 106415        SNE    0,1       ;IS BUFFER EMPTY?
   10264    414        JMP    EXIT      ;YES, EXIT MUX RETURN
   10265  35003        LDA    3,LBA.,2  ;LAST BYTE ADDRESS
   10266 136033        ADCZ#  1,3,SNC   ;END OF BUFFER?
   10267  25002        LDA    1,FBA.,2  ;YES, WRAPAROUND
   10270 125400        INC    1,1       ;BUMP OUTPUT BYTE POINTER
   10271  45005        STA    1,OBP.,2
   10272  50411        STA    2,SPCB    ;SAVE PCB PNTR
   10273   6116        GETBYTE          ;GET NEXT CHAR INTO AC2
   10274 141000        MOV    2,0
   10275  30406        LDA    2,SPCB    ;RESTORE PCB PNTR
   10276   7032        JSR    @SND.,2   ;SEND CHAR TO MUX
   10277   2403        JMP    @INTHR    ;RETURN
   10300  10402 EXIT: ISZ    INTHR
   10301   2401        JMP    @INTHR    ;EXIT RETURN TO MUX
;
   10302      0 INTHR:0
   10303      0 SPCB: 0
   10304      0 USC:  0
   10305     16 MARGN:SFTYM*2
   10306      0 WRICC:0
```

```
;********** QSTRING **********
;
   10307      0       0
   10310      0       0
   10311  54777 QSTRI:STA    3,.-1
   10312  30435       LDA    2,AATRB
   10313 112400       SUB    0,2
   10314  50773       STA    2,QSTRI-2
   10315  22772 QSTR2:LDA    0,@QSTRI-2
   10316 101112       SSP    0,0       ;NEG. CHAR. TERMINATOR?
   10317   2771       JMP    @QSTRI-1  ;YES, DONE
   10320   4404       JSR    QUP       ;NO, MOVE CHAR
   10321  10766       ISZ    QSTRI-2
   10322    773       JMP    QSTR2
;
;
;********** QUP **********
;
;
; QUE UP CHAR IN AC0 BY STORING IT IN CIRCULAR CORE BUFFER.
; IBP POINTS TO LAST CHAR STORED.
;
   10323      0       0
   10324  54777 QUP:  STA    3,.-1
   10325  32422       LDA    2,@AATRB  ;POINTER TO PCB
   10326  25004       LDA    1,IBP.,2  ;BYTE PNTR OF LAST BYTE
   10327  35003       LDA    3,LBA.,2  ;LAST BYTE IN BUFFER
   10330 136033       ADCZ#  1,3,SNC   ;WRAPAROUND?
   10331  25002       LDA    1,FBA.,2  ;YES, GET FIRST BYTE PNTR
   10332 125400       INC    1,1
   10333  45004       STA    1,IBP.,2  ;SAVE NEXT BYTE ADDRESS
   10334  34767       LDA    3,QUP-1
   10335  14747       DSZ    USC       ;REDUCE USABLE SPACE
   10336    401       JMP    .+1
   10337   6142       PUTBYTE          ;PUT BYTE INTO BUFFER
   10340   2763       JMP    @QUP-1    ;RETURN
;
;
;
   10341      0 MULCR:0       ;MULTIPLE CR MODE FLAG (0=SET)
   10342     27 ERR27:27      ;RECORD IS LOCKED ERROR
   10343     31 ERR31:31      ;ITEM TYPES DON'T MATCH ERROR
   10344     60 C60:  60      ;ASCII ZERO
   10345    117 C117: 117     ;ASCII O (OH)
   10346    561 JPROD:JMP     PROD
   10347      0 AATRB:0
```

```
;********** WRITE **********
;
;
;AC2 CONTAINS POINTER TO ICB. WRICC = # OF CHARS ALREADY HANDLED.
;IF LPT BUFFER IS OUT OF ROOM, ERROR RETURN BACK TO SYSTEM. THEN, ON
;REENTRY, WRICC > 0 MEANS IGNOR THIS # OF CHARS AS ALREADY HANDLED.
;
;
  10350      0 RTNAD:0
;
  10351  54777 WRITE:STA    3,RTNAD
  10352  21002        LDA    0,2,2    ;GET TYPE
  10353  24033        LDA    1,C11
  10354 106414        SEQ    0,1      ;ITEM TYPE = STRING?
  10355    541        JMP    WRERR    ;NO, DON'T MATCH ERROR
  10356  25003        LDA    1,3,2    ;YES, GET STRING CHAR COUNT
  10357 124513        NEGL#  1,1,SNC  ;NEG OR ZERO COUNT
  10360    472        JMP    WRIT3    ;YES, EXIT DONE
  10361  20725        LDA    0,WRICC
  10362 101005        MOV    0,0,SNR  ;ANY CHARS PREV SENT?
  10363  44542        STA    1,CHSNT  ;NO, SET FOR DONE RETURN
  10364 106400        SUB    0,1      ;YES, ADJUST FOR THEM
  10365  44533        STA    1,WRITQ  ;# OF CHARS REQUESTED FOR OUTPUT
  10366  25004        LDA    1,4,2    ;SOURCE BYTE PNTR
  10367 107000        ADD    0,1      ;ADJUST SOURCE BYTE PNTR
  10370  44531        STA    1,WRITS  ;START OF SOURCE TO OUTPUT
  10371  32756        LDA    2,@AATRB
  10372  35003        LDA    3,LBA.,2 ;LAST BYTE PNTR
  10373  25002        LDA    1,FBA.,2 ;FIRST BYTE PNTR
  10374 136400        SUB    1,3      ;BUFFER SIZE
  10375  21005        LDA    0,OBP.,2 ;CURRENT OUT BYTE PNTR
  10376  25004        LDA    1,IBP.,2 ;CURRENT IN BYTE PNTR
  10377 122023        ADCZ   1,0,SNC  ;USABLE BUFFER SPACE
  10400 163000        ADD    3,0      ;ADJUST IF OBP IS TO LEFT OF IBP
  10401  24704        LDA    1,MARGN
  10402 122423        SUBZ   1,0,SNC  ;ENOUGH SPACE FOR ZERO & MARGIN?
  10403    507        JMP    WRITX    ;NO, WAIT A BIT
  10404  40700        STA    0,USC    ;SAVE USABLE BUFFER SPACE
  10405  20644        LDA    0,INPFL
  10406 101015        SNZ    0,0      ;INIT STILL PENDING?
  10407    406        JMP    WRIT1    ;NO, CONTINUE
  10410  20513        LDA    0,INICL  ;YES, GET INIT CHARS OFFSET
  10411   4700        JSR    QSTRI    ;PUT CHARS INTO LPT BUFFER
  10412 102400        SUB    0,0
  10413  40636        STA    0,INPFL  ;CLEAR INIT PENDING FLAG
  10414  40725        STA    0,MULCR  ;RESET MULTIPLE CR MODE
```

```
- PAGE 5 -

;WRIT1 TRANSFERS CHARS FROM THE USERS BUFFER TO LPT'S BUFFER
;
;
   10415  20667 WRIT1:LDA    0,USC
   10416 101112       SSP    0,0       ;ANY USABLE SPACE LEFT?
   10417    473       JMP    WRITX     ;NO, WAIT A BIT
   10420  24501       LDA    1,WRITS   ;YES, FETCH SOURCE POINTER
   10421   6151       XGETBYTE          ;GET CHAR INTO AC2
   10422  20051       LDA    0,C177
   10423 143405       AND    2,0,SNR   ;IS CHAR A NULL?
   10424    426       JMP    WRIT3     ;YES, END OF STRING
   10425  24517       LDA    1,WRITL   ;CHAR LOW RANGE
   10426  34517       LDA    3,WRITH   ;CHAR HIGH RANGE
   10427 122432       SUBZ#  1,0,SZC   ;SKIP IF CHAR < LOW RANGE
   10430 162032       ADCZ#  3,0,SZC   ;SKIP IF CHAR <= HIGH RANGE
   10431    430       JMP    WRIT6     ;NO, GO CHECK IT
   10432  40707       STA    0,MULCR   ;YES, RESET MULTIPLE CR MODE
   10433  34714       LDA    3,AATRB
   10434  25776       LDA    1,EXOFF,3
   10435 125213       SKO    1,1       ;EXCHANGE O (OH) & ZERO?
   10436    407       JMP    WRIT4     ;NO
   10437  24706       LDA    1,C117    ;YES
   10440  30704       LDA    2,C60
   10441 106415       SNE    0,1       ;IS CHAR O (OH)?
   10442 145001       MOV    2,1,SKP   ;YES, SEND 0 (ZERO)
   10443 112415       SNE    0,2       ;IS CHAR ZERO?
   10444 121000       MOV    1,0       ;YES, SEND O (OH) INSTEAD
   10445   4657 WRIT4:JSR    QUP       ;SEND CHAR TO LPT BUFFER
   10446  10640 WRIT2:ISZ    WRICC
   10447  10452       ISZ    WRITS
   10450  14450       DSZ    WRITQ     ;DONE WITH REQUESTED STRING?
   10451    744       JMP    WRIT1     ;NO, LOOP BACK
   10452   4455 WRIT3:JSR    PROD      ;"KICK" THE PRINTER
   10453 102400       SUB    0,0
   10454  40632       STA    0,WRICC   ;CLEAR CHARS SENT COUNT
   10455  20450       LDA    0,CHSNT   ;GET CHARS SENT COUNT
   10456  34672       LDA    3,RTNAD
   10457   1401       JMP    1,3       ;GOOD RETURN TO SYSTEM
   10460    631 JQSTR:JMP    QSTRI
;
   10461  24037 WRIT6:LDA    1,C15     ;ENTRY NOT MIDRANGE
   10462 106414       SEQ    0,1       ;IS CHAR A CR?
   10463    413       JMP    WRIT5     ;NO, TEST FOR SPECIAL CHAR
   10464  30655       LDA    2,MULCR
   10465 151004       MOV    2,2,SZR   ;MULTIPLE CR MODE SET?
   10466    403       JMP    WRIT7     ;NO, GO SET IT
   10467  20034       LDA    0,C12     ;GET LF CHAR
   10470    416       JMP    WRIT8     ;QUEUE IT UP WITH DELAY STRING
;
   10471 152400 WRIT7:SUB    2,2
   10472  50647       STA    2,MULCR   ;SET MULTIPLE CR MODE
   10473  20427       LDA    0,WRCRL
   10474   4764       JSR    JQSTR     ;SEND CR CHARS INSTEAD OF CR
   10475    751       JMP    WRIT2     ;YES, NOT FOUND - IGNORE IT
```

```
   10476   4450 WRIT5:JSR    WRITP    ;AC3=PNTR TO SPECIAL CHAR LIST
   10477  25400       LDA    1,0,3
   10500 125112       SSP    1,1      ;END OF LIST (-1)?
   10501    745       JMP    WRIT2    ;YES, NOT FOUND - IGNORE IT
   10502 122415       SNE    1,0      ;CHAR IN LIST?
   10503    403       JMP    WRIT8    ;YES, PASS SPECIAL CHAR TO LPT
   10504 175400       INC    3,3      ;NO, KEEP CHECKING
   10505    772       JMP    WRIT5+1
;
   10506   4616 WRIT8:JSR    QUP      ;SEND CHAR TO LPT BUFFER
   10507  20415       LDA    0,NULST
   10510   4750       JSR    JQSTR    ;SEND NULLS (DELAY)
   10511    735       JMP    WRIT2    ;CONTINUE
;
   10512   4415 WRITX:JSR    PROD     ;"KICK" THE PRINTER
   10513  24504       LDA    1,WRITD  ;LOCKED RETURN DELAY
   10514  34626       LDA    3,ERR27  ;"RECORD IS LOCKED" ERROR
   10515   2633       JMP    @RTNAD   ;ERROR RETURN
;
   10516  34625 WRERR:LDA    3,ERR31  ;"ITEM TYPES DON'T MATCH" ERROR
   10517   2631       JMP    @RTNAD
;
   10520      0 WRITQ:0               ;# OF CHARS REQUESTED IN WRITE
   10521      0 WRITS:0               ;BYTE ADDRESS OF SOURCE STRING
   10522     43 WRCRL:ATRIB-WRICR     ;OFFSET TO CR CHAR LIST
   10523     33 INICL:ATRIB-INITZ     ;OFFSET TO INIT CHAR LIST
   10524     13 NULST:ATRIB-NULLS     ;OFFSET TO NULLS LIST
   10525      0 CHSNT:0               ;CHARACTERS SENT CELL
;
;
;********** PROD ROUTINE **********
;
;PROD STIMULATES THE FIRST INTERUPT FROM THE LPT IN ORDER TO GET
;OUTPUT GOING. ALSO TO "KICK" LPTIN THE EVENT IT FAILS TO
;COMPLETE PRINTING, YET REMAINS READY.
;
;
   10526      0       0
   10527  54777 PROD: STA    3,.-1
   10530  32617       LDA    2,@AATRB ;PNTR TO PCB FOR LPT
   10531  35005       LDA    3,OBP.,2
   10532  25004       LDA    1,IBP.,2
   10533 136415       SUB#   1,3,SNR  ;BUFFER EMPTY?
   10534   2772       JMP    @PROD-1  ;YES, EXIT
   10535  21001       LDA    0,OCW.,2
   10536 101102       MOVL   0,0,SZC  ;IS MUX BUSY?
   10537   2767       JMP    @PROD-1  ;YES, WAIT FOR INTERRUPT
   10540 102400       SUB    0,0      ;SEND A NULL
   10541  32606       LDA    2,@AATRB ;SET AC2=PCB FOR MUX
   10542   7032       JSR    @SND.,2  ;SEND CHARACTER TO MUX
   10543   2763       JMP    @PROD-1
```

```
- PAGE 7 -

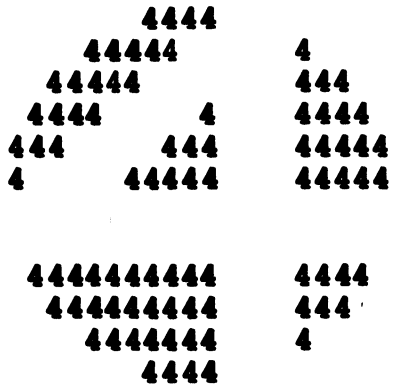
;********** TABLES AND WORKING STORAGE **********
;
;
   10544     40 WRITL:40      ;LOWEST NON-SPECIAL ASCII CHAR
   10545    174 WRITH:174     ;HIGHEST NON-SPECIAL ASCII CHAR
;
   10546   5400 WRITP:JSR     0,3       ;GENERATE PNTR TO FOLLOWING LIST
   10547     14               14
   10550      0               0
   10551     12               12
   10552      0               0
   10553 177777               -1
   10554 177777               -1
   10555 177777               -1
   10556 177777               -1
;
;SEND THIS STRING IN PLACE OF CR
;
   10557     15 WRICR:15      ;CR CHAR LIST
   10560      0               0
   10561     12               12
   10562      0               0
   10563 177777               -1
   10564 177777               -1
   10565 177777               -1
   10566 177777               -1
;
              0               .DMR WRICZ=JMP WRICR+SFTYM+1-.;CR CHAR OVERFLOW TEST
;
;THE INIT CHARS ARE OUTPUT WHEN LPT IS OPENED
;
   10567     15 INITZ:15
   10570      0               0
   10571     14               14
   10572      0               0
   10573 177777               -1
   10574 177777               -1
   10575 177777               -1
   10576 177777               -1
;
              0               .DMR INIZZ= JMP INITZ+SFTYM+1-.;INITZ OVERFLOW TEST
```

```
- PAGE 8 -

;THE FINIZ CHARS ARE OUTPUT WHEN LPT IS CLOSED
;
   10577      15 FINIZ:15      ;CR FLUSHES THE LPT BUFFER
   10600       0        0
   10601  177777        -1
   10602  177777        -1
   10603  177777        -1
   10604  177777        -1
   10605  177777        -1
   10606  177777        -1
;
               0        .DMR FINZZ= JMP FINIZ+SFTYM+1-.;FINIZ OVFLO TEST
;
;
;TIMING STRING SENT AFTER ALL SPECIAL CHARS
;
   10607       0 NULLS:0
   10610       0        0
   10611       0        0
   10612  177777        -1
   10613  177777        -1
   10614  177777        -1
   10615  177777        -1
   10616  177777        -1
;
               0        .DMR NULLZ= JMP NULLS+SFTYM+1-.;OVFLO TEST
;
   10617       3 WRITD:3       ;LOCKED RETURN DELAY, ADJUST FOR MAX LPT SPEED
;
;
   10620       0 EXCHF:0
;
;
   10621       2 PORTN:2       ;LPT ASSIGNED TO PORT # +1 (DEFAULT IS PORT 1)
   10622       0 ATRIB:0       ;PCB FURNISHED
   10623       0        0
   10624       0        0
   10625  177777        -1      ;LINKAGE POINTER TO TERMINATOR
   10626  177777        -1      ;NO PORT DEFINATION TABLE
;
         177776        EXOFF= EXCHF-ATRIB ;OFFSET TO EXCHANGE FLAG
         177777        PTOFF= PORTN-ATRIB ;OFFSET TO PORT NUMBER
;
;
                       .END
```

AATRB	10347	AD1	30	AD2	31	AI1	26	AI2	27
ATRIB	10622	BBA	15	BINDI	6121	BINMU	6122	BPI	16
BSACF	76	BUMPU	6123	C10	32	C100	50	C1000	66
C11	33	C117	10345	C12	34	C13	35	C14	36
C15	37	C16	40	C160	174	C163	175	C166	176
C17	41	C170K	73	C171	177	C177	51	C1777	67
C2	2	C20	42	C200	52	C2000	70	C205	53
C20K	10226	C215	54	C240	55	C244	56	C260	57
C271	60	C3	3	C300	61	C334	62	C37	43
C377	63	C4	22	C40	44	C400	64	C4000	71
C5	23	C6	24	C60	10344	C600	115	C7	25
C77	45	C774C	72	C777	65	CALL	6100	CHANN	6105
CHSNT	10525	CM400	21	DA	160	DAC	164	DAS	165
DB	166	DBC	172	DBS	173	DECIM	6124	DQUEU	6104
ERR27	10342	ERR31	10343	ERR43	10234	ERRF	77	ESCF	74
EXCHF	10620	EXIT	10300	EXOFF	177776	FINCL	10250	FINDL	6131
FINI2	10243	FINIS	10235	FINIZ	10577	FIX	6127	FLAGC	6101
FLOAT	6130	FREEN	6106	GETBY	6116	INBYT	6117	INERR	10232
INICL	10523	INIT	10205	INIT1	10211	INITZ	10567	INPFL	10251
INSTB	6120	INTH	10253	INTHR	10302	ISA2D	6132	ISA2L	6133
JPROD	10346	JQSTR	10460	LOADD	6135	MARGN	10305	MULCR	10341
NULLS	10607	NULST	10524	OUTBY	6146	OUTTE	6136	PORTN	10621
PROD	10527	PTOFF	177777	PUTBY	6142	QCHAR	6102	QSTR2	10315
QSTRI	10311	QUEUE	6103	QUP	10324	READB	6140	RTL	75
RTNAD	10350	RTP	7	RUP	5	RUS	6	SFTYM	7
SPCB	10303	STINP	6144	STORD	6143	STOUT	6145	TRAPF	6126
USC	10304	WRCRL	10522	WRERR	10516	WRICC	10306	WRICR	10557
WRIT1	10415	WRIT2	10446	WRIT3	10452	WRIT4	10445	WRIT5	10476
WRIT6	10461	WRIT7	10471	WRIT8	10506	WRITB	6150	WRITD	10617
WRITE	10351	WRITH	10545	WRITL	10544	WRITP	10546	WRITQ	10520
WRITS	10521	WRITX	10512	XGETB	6151	XPUTB	6152	.ABA	14
.ACBY	116	.ACIB	117	.ACSB	120	.ATRB	10201	.BDIV	121
.BMUL	122	.BPS	114	.BSA	10	.BUMP	123	.CALL	100
.DA	174	.DA3	175	.DB	176	.DB3	177	.DEC	124
.FALT	126	.FIX	127	.FLOT	130	.FLUT	131	.HBA	11
.HXA	12	.IA2D	132	.IA2L	133	.INFO	115	.INTR	134
.LCM	111	.LODA	135	.MSG	136	.NRET	137	.PCA	113
.RBLK	140	.SRET	141	.SSA	13	.STBY	142	.STDA	143
.STI	144	.STO	145	.STOB	146	.STPL	147	.WBLK	150
.XACB	151	.XSTB	152						

POINT 4 Data Corporation



T E C H N I C A L
M E M O R A N D U M

TO: IRIS LCM USERS

FROM: IRIS CUSTOMER SUPPORT

DATE: May 14, 1982

SUBJ: LCM DEFAULT PARAMETERS FILE

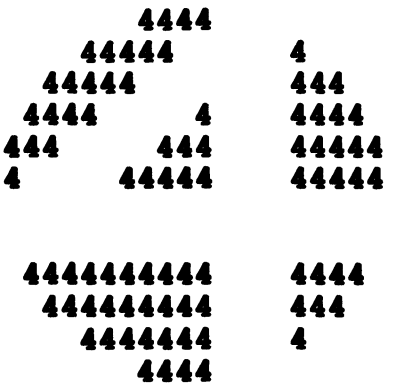
The LCM parameters file (LCM.PARAMS) is shipped with the following configuration:

```
*HAMMING
*CERRCOUNT
*SIZE=1024
@1-10
0/DNESTBLKS
0/INDEX,ACCOUNTS,BASIC,RUN,RUNMAT:C
```

Our experience shows that an LCM set up with this configuration delivers good performance. As indicated in the LCM documentation, better performance may be achieved by tuning the LCM to reflect individual system use.

If the default LCM.PARAMS file is used, it must be modified to reflect the size of the LCM (*SIZE=n), and the active files for the interactive ports on your system (@n-m).

POINT 4 Data Corporation



T E C H N I C A L
M E M O R A N D U M

TO: All IRIS Users

FROM: IRIS Customer Support

DATE: June 30, 1982

SUBJ: TURNING ECHO ON/OFF FROM A BASIC PROGRAM

The following program shows how echo can be turned on or off from a BASIC program. It may be run from any account.

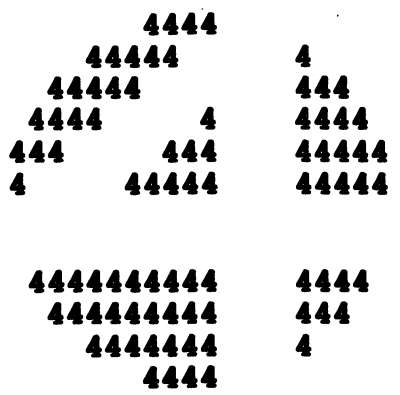
ECHO

```
10 DIM A$[20]
20 CHAIN "BASIC\215\\205\\215\30 RUN"

30 INPUT "TYPE SOMETHING, THEN CR (ECHO IS DISABLED):  "A$
40 REM THIS WILL NOT BE ECHOED AS YOU TYPE IT
50 PRINT "\215\\215\YOU TYPED:  ";A$;
60 PRINT "     # OF CHARS ENTERED = "; LEN (A$)
70 CHAIN "BASIC\215\\205\\215\80 RUN\205\"

80 INPUT "TYPE SOMETHING (ECHO ENABLED AGAIN):  "A$
90 PRINT "\215\\215\ALL DONE"
100 CHAIN ""
```

POINT 4 Data Corporation



TECHNICAL MEMORANDUM

TO: All IRIS Users

FROM: IRIS Customer Support

DATE: June 30, 1982

SUBJ: COPYING CONTIGUOUS FILES

To copy a contiguous or indexed contiguous file, the number of records (R) and the number of words per record (W) must be specified, e.g.

```
COPY [R:W] lu/destination filename=lu/source filename
```

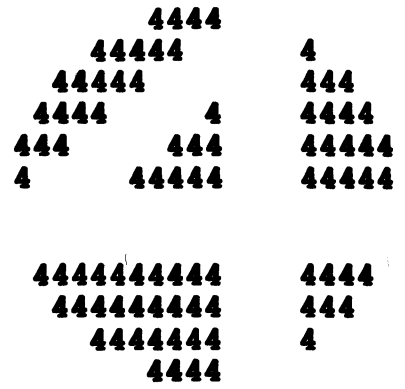
The following program shows how to determine Records:Words.

NOTE

Program must be run from the Manager Account (0,1) only!

```
10 LET N=3
20 OPEN #N,"CONTIGFILE"
30 REM GIVEN A CONTIGUOUS FILE OPEN ON CHANEL N
40 LET R= CHF (N)
50 LET W= SPC ( SPC ( SPC (5+32768)+21+32768)+N*8+5+32768)
60 PRINT  USING "R:W = #####: #####";R,W
70 END
```

POINT 4 Data Corporation



TECHNICAL MEMORANDUM

TO: All IRIS Users (and LCM Users)

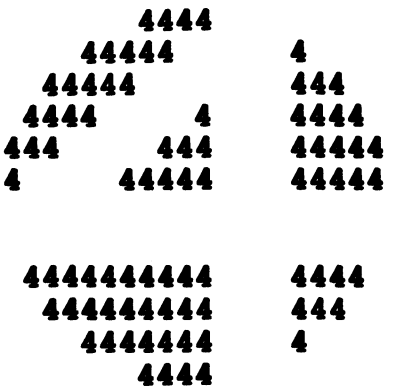
FROM: IRIS Customer Support

DATE: June 30, 1982

SUBJ: CAUSE OF ERROR 38s

Any BASIC program (including LCM CONFIGURE) that uses CALL 95 must be on the Manager's Account (0,1) and must be saved at protection 77. If either case is not true, an error 38 will occur.

POINT 4 Data Corporation



TECHNICAL MEMORANDUM

TO: MARK 3 BASF PSS Users

FROM: IRIS Customer Support

DATE: June 30, 1982

SUBJ: BASF PSS CUSTOMER CHECKOUT PROCEDURE

When you receive a BASF PSS, we recommend that you follow the procedure given below to ensure that everything is working properly.

PROCEDURE:

1. FORMAT the BASF disc drive
2. Use RESTORE to copy the operating system (sent to you by POINT 4) from archive tape to your disc
3. IPL
4. SHUTDOWN to DISCUTILITY
5. Use SAVE to copy operating system from BASF to scratch tape
6. FORMAT the BASF disc drive
7. Use RESTORE to copy the tape you saved in step 5, to the BASF
8. IPL

APPENDICES

Manager Manual Addenda #1

Corrections for IRIS 7.3 Manager Reference Manual
EDS 1018-11 5 SEP 78

Pg. 2-12; Paragraph starting "606 (Not currently used).": Replace this with the following: "606 NDPF No Dirty Page Flag. If set non-zero, some of the performance advantage of the buffer pool will be lost. However, by forcing all block writes directly to disc, the reliability of data on the disc is assured, even if memory is cleared."

Pg. 2-13; Paragraph starting "610 TOPW TOP Word of ...": Insert at end of paragraph: "Do not set TOPW above 77777 unless your CPU and all disc controllers on your system correctly use a 16-bit memory address."

Pg. 2-24; Paragragh starting "3) Select a suitable ..."; Line 4 starting "and do an IPL ...": Replace this line by the following: "Also change the LBSA cell as described on 2-10. Note that MBUS for 7.3 currently is 20600. SHUTDOWN the system and do an IPL to make these values effective."

Pg. 2-28; Paragraph starting "11 Same as mode 10 ...": Replace this whole paragraph by the following: "11 Same as mode 10, but log-on is allowed if any entry in the whole table both matches and allows log-on. In all other modes, scan stops with the first match."

Pg. 4-4; Paragraph starting "8) Do an IPL by ...": Replace the first sentence by the following: "8) Exit from DSP, SHUTDOWN the system, and do a new IPL."

Pg. 4-5; Paragraph 1 starting "4.6 How to Replace SCOPE, ..."; Sentence in line 4 starting: "To do this, RESET and ...": Replace this whole sentence by the following: "To do this, exit from PLOAD by CTRL C, SHUTDOWN the system, and do a new IPL."

Pg. A6-6; Paragraph 1 starting "$ALU (DCC ALU ...": Add the following line at the end, just before the line starting "TTY or TTY50 ...": "Remember to set the total number of ports in the word just before ATRIB."

Pg. A6-6; Paragragh starting "$PHA (Phantom ...": Insert at the end: "Set the PCW word to 2000. Also, do not change the word just before ATRIB."

MANAGER MANUAL ADDENDA #2

Corrections for IRIS 7.3 Manager Reference Manual

EDS 1018-11 5 Sept. 78

Pg. 2-3; Paragraph starting "4) A real time clock."; Delete this paragraph. Insert the following:

4) A real time clock. This may be supplied by a standard Real Time Clock (with device code 14, supplying interrupts at 10 Hz) or any EDSI multiplexer.

Pg. 8-10; Eighth line on page: "8.8 More on INSTALL"; insert after this line the following paragraph:

"Note: The following section discusses logical unit formatting, not disc formatting. INSTALL assumes that a stand-alone format program, which must be supplied by the disc controller vendor (not EDSI), has already been run."

Pg. 8-11; Last sentence in first paragraph ending: "a new (empty) Logical Unit." ; insert after this sentence the following: "REHASH must be the first thing you run on the Logical Unit which you have just INSTALL AND CLEARed."

Pg. A1-14; Last sentence on page starting: "Return is non-skip...";insert after this sentence the following: "(see also READ CONTIGUOUS)."

Pg. A1-20; Sentence in Line 7 from bottom of page starting: "Note: for toggle,...": insert before this sentence the following: "FLAG-CHANGE disables interrupts and returns with interrupts enabled."

Pg. A6-1; insert at bottom of page: "The following are not currently supported under IRIS:

card readers
papertape through multiplexer port"

Pg. A6-4; insert at bottom of page:
"Note: An interactive port on the Mighty-Mux is always assumed to be ready to receive characters. If it is desired to slave a printer, cassette, or floppy through a CRT, enough delays must be put at the end of line and a low enough baud rate used to prevent loss of characters."

Insert this section after pg. 8-16. This supersedes CONVERT memo of 17 April 1977.

8.15 How to CONVERT an R4 Logical Unit to R7.3.

Any IRIS user updating a system from R4.3 to R7.3, or for that matter from any IRIS or revision R7.0 or earlier to any revision R7.1 or later, must convert each Logical Unit before using it under the newer system. This is because all Real Disc Addresses are sequential starting with R7.1. A CONVERT processor is supplied by EDS to perform this conversion. To update to IRIS R7.1 or R7.3 or later, follow these steps carefully:

1. Back up all Logical Units.

2. If files on the old Logical Unit zero are to be carried forward, than either:

    a) INSTALL a copy of it under the old system, changing its Logical Unit number to any unused non-zero unit number, or

    b) COPY all files to be carried forward to a different Logical Unit.

3. DUMP all BASIC program to text files, and KILL the SAVEd version of all BASIC programs.

4. Do a SysGen of the new IRIS on a scratch pack. It is strongly recommended that Logical Unit zero be only large enough for the system itself and that all user and application files be on other Logical Units; this is to make it easier to do later updates.

5. Be sure that the old disc address conversion constants are in the new CONFIG file in word 7 of each partition table entry in the form LRC*100+LRT where LRC is the Logical-to-Real Cylinder conversion factor, LRT is the Logical-to-Real Track conversion factor, and 100 is an octal constant. For example, if LRC was 40 and LRT was 20 on the old system, then enter 4020 octal in location 1423 of CONFIG for partition 0.1, in location 1433 for partition 0.2, etc. The old values of LRT and LRC are given in the red 7.3 CONFIG LAYOUT book. GUIDE will help you set up the Logical Unit tables in CONFIG for this conversion.

6. Mount an old Logical Unit, and give a system command of the form

    # CONVERT d.p

    where d.p is the partition as defined for INSTALL. DO NOT INSTALL THE UNIT BEFORE CONVERSION! If a "conversion not needed" message is printed, this is because the conversion constants were already such that the Real Disc Addresses were sequential, and conversion is not required. That is, conversion is required if the new 7.3 LRT and LRC do not equal the old R4 LRT and LRC.

7. The unit will be converted to the new disc addressing method. It should take about three times as long to convert as it takes to INSTALL the same unit. Repeat step 6 for each old Logical Unit.

The Logical Units may now be INSTALLED as normal R7.3 logical units. If the old LRT and LRC do not equal the new LRT and LRC, then the conversion will prevent the Logical Unit from being used under R4 ever again. Only if "conversion not needed," is a Logical Unit usable under either R4 or R7.3 interchangeably. Beware: Do not try a second time to convert a Logical Unit which has already been converted to 7.3 once. This will turn most of the file data to garbage.

Also, if for any reason, CONVERT fails to run to completion, then the Logical Unit is lost and must be restored from the back up.

After the Logical Unit is CONVERTed, it must then be INSTALLed and REHASHed to run under 7.3. (see 8.16 below)

8.16 How to REHASH a Logical Unit:

1. CHANGE any filename on that logical unit so that it does not start with a "$" (esp. if this was LU/0).

2. Backup the logical unit.

3. Log onto the manager account.

4. INSTALL the logical unit.

5. Make sure that there are enough available blocks and that the manager is allotted enough blocks to build a temporary file on that logical unit the size of its INDEX.

6. Enter the system command

REHASH

which will ask

LOGICAL UNIT TO REHASH?

7. If REHASH traps or abnormally terminates, do not use that logical unit any more. Restore it from the back up and REHASH again.

8. REHASH can be run periodically to speed up all OPEN and CLOSE operations on files on that logical unit, (esp. if files are often created and deleted).

9. REHASH must be run
   a) after INSTALL AND CLEAR
   b) after CONVERT
   c) if the logical unit was created under R7.2 or R4.

MANAGER MANUAL ADDENDA
#3

Corrections and additions for the IRIS Manager Manager Addenda

Change
Ch. 1.3, Page 1-4, paragraph 3

Reads: ", an Accounts file whose header is in Real Disc Address three, and a DMAP"

Should Be: ", an Accounts file whose header is in Real Disc Address three, and whose first block is in Real Disc Address five, and a DMAP"

Manager Manual Addenda
#3

Corrections and Additions for the IRIS Manager Reference MANUAL
EDS 1018-11 5 Sept. 78

Change
Ch. 2.4.1, Page 2-13, paragraph 1

Reads: "Any core available above 77777
Octal will be used only for user partitions."

Change to: "Any core available above 77777 octal
will be used only for user partitions and
Buffer Pooling."

MANAGER MANUAL ADDENDA
#3

Corrections and Additions for the Manager Reference Manual
EDS 1018-11 5 Sept. 78

Change
Ch 3.2, Page 3-6, Paragraph 1

Reads: "(SGR, SGE, SLS, SLE, SEQ, SNE, SKZ, SNZ, SSP, SSN, SGZ, SZN, SKE, and SKO)"

Should Be: "(SGR, SGE, SLS, SLE, SEQ, SNE, SKZ, SNZ, SSP, SSN, SGZ, SNP, SKE and SKO)"

MANAGER MANUAL ADDENDA
#3

Corrections and additions for the IRIS Manager Reference Manual
EDS 1018-11 5 Sept. 78

Addition
Ch. 4.1, Page 4-2

Insert after paragraph 2, "If PLOAD prints RDR OK?"

PLOAD (with type 77003) or COPY (filename *A) can be used to load stand-alone papertape diagnostics under IRIS. See section 8.4 to run them. Some papertapes, if not supplied by Point 4 Data Corporation, may use a punch format which will prevent them from being loaded under IRIS even though they can be loaded by our binary loader.

MANAGER MANUAL ADDENDA
#3

Corrections and additions for the IRIS Manager Reference Manual
EDS 1018-11 5 Sept. 78

Change
Page 10.3, Paragraph 2

Reads: "Discsub numbers 120 through 127 octal have been set aside for customers to assign to subroutines for their own use only."

Change to: "Discsub numbers 130 through 137 octal are reserved for customer use."

Change
Page 11-25, Paragraph 5

Reads: If A0 = 0 then nothing is stored in the IO buffer."

Change to: "If A0 = -2 then nothing is stored in the IO buffer."

Addition
CH 8. 10, Page 8-12

The system manager can declare a port type by giving the command

PORT p TYPE n

Where p is a logical system port number and n is the port type of an active terminal translation module. Refer to the IRIS Peripherals Handbook to find the port type. An invalid port number, inactive module, or inactive $TERMS system driver will give an error message.

MANAGER MANUAL ADDENDA
#3

Corrections and additions for the IRIS Manager Reference Manual
EDS 1018-11 5 Sept. 78

Addition
CH 10.4, Page 10-7

Beginning with IRIS R7.4, the following BASIC call numbers are in use or reserved:

1
3
22
24
80 - 99

Addition
CH 10.2, Page 10-4

The IRIS 7.4 DISCSUBS file has two reserved areas. They are;

Loc. 40400 - 41777 reserved for future IRIS use

Loc. 42000 - 43777 reserved for customer use

Please use the area starting at location 42000 to add your discsubs. If you need more room, append blocks starting at location 44000. Remember, after appending a block to DISCSUBS you must run CLEANUP.

MANAGER MANUAL ADDENDA
#3

Corrections and additions for the IRIS Manager Reference Manual
EDS 1018-11 5 Sept. 78

Change
Ch 11.8, Page 11-19, paragraph 2

Reads: "NSECT indicates the number of sectors (number of blocks per track), not to exceed 16 (octal 20) sectors. NTRK indicates the number of tracks per clinder (number

Should be: "NSECT indicates the number of sectors (number of blocks per track), not to exceed 16 (octal 20) sectors. The minimum value is six. NTRK indicates the number of tracks per cylinder (number

Change
Ch 11.8, Page 11-19, paragraph 2

Reads: "Discs having more than 16 sectors must be specified otherwise;

Should be: "Discs having more than 16 sectors, or less than 6 sectors, must be specified otherwise;

MANAGER MANUAL ADDENDA
#3

Corrections and additions for the IRIS Manager Reference Manual
EDS 1018-11 5 Sept. 78

Addition
Ch. 11.10, Page 11-23

Add to paragraph 2, "Each terminal..."

"Terminal Type Codes 120 through 144 are reserved for customer use."

MANAGER MANUAL ADDENDA
#3

Changes and additions for the Manager Reference Manual
EDS 1018-11 5 Sept. 78

Change
Ch. 11.10, Page 11-25, paragraph 2

Reads: "If the search fails, an "↑" is output instead of the Byte.

Should be: "If the search fails, a null, ASCII 200, is output instead of the Byte.

Change
Ch. 11.10, Page 11-25, paragraph 2

Reads: "If the search succeeds, the action depends on the value of the P-Bit. If P=0, the 7 Bit TRANS is output with the high order eighth bit unconditionally set."

Should be: "If the search succeeds, action depends on the value of the P-Bit and the E-Bit.

E	P	
(0 or 1)	1	Jump to procedure starting at offset = (TRANS + E*200) from table entry where match is found.
1	0	Store TRANS in output buffer
0	0	Store terminal's ESC Code in output buffer followed by TRANS

If P=0, the 7-Bit TRANS is output with the high-order eighth Bit unconditionally set."

MANAGER MANUAL ADDENDA
#3

Corrections and additions for the IRIS Manager Reference Manual
EDS 1018-11 5 Sept. 78

Addition
11.11 How to Install a Terminal Translation Module

The system manager activates a terminal translation module as follows:

Obtain the terminal translation module file name for the terminal from the IRIS Peripherals Handbook.

Enable the selected terminal translation module as a dollar sign file, TYPE 77001.

Enable the system driver $TERMS as a dollar sign file.

A terminal translation module, including the ability to use its corresponding basic terminal control mnemonics, is active at a port when the port is linked to it. To have the system automatically link one or more ports at IPL:

Obtain the module's terminal type code (TTC) from the IRIS Peripherals Handbook.

Locate each port's RDE cell in the Port's Device Driver File. Refer to Section 11.1, "Interactive and Peripheral Device Drivers", if necessary.

Use DSP to store the TTC in the lower right hand byte of each selected port's RDE cell.
(Note: The upper left hand byte is reserved by the system for the port's return delay. This delay is still valid after storing a TTC.)

Shutdown and RE-IPL the system.

To link or change linkage between a port and a terminal translation module after an IPL.

Obtain the module's port type from the IRIS Peripherals Handbook.

From any account, declare a port type. I.E.

PORT TYPE n

See the IRIS User Manual, Section 1.4.

The System Manager can direct a port declaration to any interactive port using the port command:

PORT p TYPE n

See the IRIS Manager Manual Section 8.10.

You cannot pass characters less than 200 octal directly to your screen with a terminal translation module linked to your port. Declaring

PORT TYPE 0

Removes linkage and allows characters to pass as they are. All ports are type zero until linked to an enabled module.

Terminal translation modules are reentrant and shared by all ports linked to them. Therefore, link any number of ports to a single module.

The system accepts up to 15 enabled dollar sign modules. Each enabled module must have a unique terminal type code (TTC).

User Manual Addenda #1

Corrections for IRIS 7.3 User Reference Manual
EDS 1017-11 5 SEP 78

Pg. 1-11; BASIC program starting "400 LET D= ..."; Line 4 starting "430 IF SPC(D+24) ...": Replace this line by "430 IF SPC(D+30)<>1024 GOTO 450 ! PCW. = 36".

USER MANUAL ADDENDA
#2

Corrections and additions for the IRIS User Reference Manual
EDS 1017-11 5 Sept. 78

CH 1.4, Page 1-5
Addition

PORT TYPE n

Links the active terminal translation module whose port type is n to the port where the command is given. Refer to the IRIS Peripherals Handbook to find the correct port type. Declaring the port type of an inactive module, or inactive $TERMS system driver, will give an error message.

USER MANUAL ADDENDA
#2

Corrections and additions for the IRIS User Reference Manual
EDS 1017-11 5 Sept. 78

Addition
Ch. 2.9, pg. 2-11

At top of page, insert

For stand-alone programs, see Manager Manual Section 4.1.

USER MANUAL ADDENDA
#2

Corrections and additions for the IRIS User Reference Manual
EDS 1017-11 5 Sept. 78

Addition
6.13 New features of OEM BASIC

Introduction

OEM BASIC has become the standard BASIC on all IRIS systems since 7.4. It was previously released only on a limited basis. It provides a means for protection of proprietary applications by OEM's. This protection is effected by removal of the ability to LIST or DUMP PROTECTed BASIC programs. Programs SAVEd under old BASIC are fully upward compatible to OEM BASIC. BASIC programs are protected by using a new processor, PROTECT. A new processor, VERIFY, is provided to aid in the mainenance of BASIC programs.

Compatibility

Old SAVEd BASIC files (file type 2) are fully upward compatible with OEM BASIC. However, OEM BASIC is NOT downward compatible; it uses a different BASIC program format than that for old BASIC. OEM BASIC knows about the format differences between itself and old BASIC and is therefore able to run programs that have the old SAVEd BASIC format. Old BASIC, however, does not know of these format differences and consequently cannot run programs SAVEd or PROTECTed in OEM format. This in itself is a mild form of protection in that OEM BASIC programs cannot be run under an old BASIC system. NOTE: Trying to run an OEM BASIC program on an old BASIC system may cause the system to crash. Consequently, old BASIC and OEM BASIC should not be co-resident on the same system. Transporting unPROTECTed programs from an OEM BASIC system to an old BASIC system must be done by doing a DUMP of the programs under OEM BASIC and then a LOAD of the program's text under old BASIC.

Protection of Programs

To PROTECT programs, the PROTECT processor is used. PROTECT changes the program to a non-listable form. PROTECT is used in exactly the same way that SAVE is. EXAMPLE:

```
#BASIC
100 PRINT "THIS WILL BE SAVED IN PROTECTED FORM."
 \       [control-C]
#PROTECT BROWNMOTIE
```

The program shown is SAVEd in PROTECTed form under the filename "BROWNMOTIE". The "!" and "<>" options that are available when using SAVE are also available with PROTECT. Once PROTECTed, a program cannot be unPROTECTed. So, don't PROTECT your only copy of the program. PROTECT first makes the program unlistable, then automatically tries to SAVE it in that format. If the SAVE part fails (e.g., filename already in use), then just SAVE it with a correct filename.

Lines may be entered or modified or a text file containing lines may be LOADed into PROTECTed programs just as into non-protected programs. These changes may be reSAVED but the whole program will still remain PROTECTed. A "DELETE" with no arguments or a "NEW" will clear the program area and exit from list protect mode.

Check Codes

The processors SAVE, PROTECT, and VERIFY that are provided with OEM BASIC all display a Check Code upon completion. Any change to the BASIC program will randomly change the program's Check Code. The sole function of the VERIFY processor is to display this code. The purpose of the code is to detect if any program changes have been made, authorized or not, since the last Check Code was recorded by the programmer. This is very useful when patches are put into a PROTECTed program to verify that they were correctly done. It may also be used to detect if a customer has done some unauthorized modification of a program.

Note that the checksum depends on the order in which the lines are entered. For two programs to have the same checksum, all lines and patches must be entered in the same sequence.

USER MANUAL ADDENDA
#2

Corrections and additions for the IRIS User Reference Manual
EDS 1017-11 5 Sept. 78

Delete Section 4.3 "How to EXECUTE a program" (EXECUTE is no longer a product)

Delete Section 6.12 "How to LOAD a BASIC program from cards"

Delete Appendix 4 "HOLLERITH and DRATSAB CODES"

Delete Appendix 5 "EDSI BASIC CARD PROGRAMMER"

Change
Ch. 12.9, page 12-24, paragraph 5

Reads:
m=7 Used to re-organize a directory. Directory d is optimized for effecient packing so that more keys may be inserted (see Section 12.10).

change to:
m=7 Used to re-organize a directory. Directory d is optimized for efficient packing so that more keys may be inserted (see Section 12.10).

POINT 4 DATA CORPORATION
2569 McCabe Way / Irvine, California 92714 / (714) 754-4114