CONTENTS

subject: **3B2/300 UNIX* System V Release 2 Devtools Guide**

date: **February 22, 1985**

from: **M. H. Murnighan**
CU T2CU132130
C15 x6165

Development tools (DEVTOOLS) are an unsupported utility package for the 3B2/300. DEVTOOLS were put together for internal support during the development stage. These utilities are of great use during development, specifically for the addition of foreign disk drives.

DEVTOOLS contains 9 programs:

1. Formflop: formats floppy disks.
2. Forhard: formats hard disks.
3. Ddhs: disk to disk high speed copies.
4. Dd: disk to disk copy by sectors.
5. D-m: disk to memory copies.
6. Defect: defect table builder.
7. Fixdisk: writes sanity track.
8. Duf: duplicate floppy disk.
9. Chgboot: Changes the default booting information.

1. DEVTOOLS USAGE

DEVTOOLS is a firmware tool, it will control the 3B2 when used in place of UNIX. For this reason you must be in the firmware mode to boot up on DEVTOOLS. The following steps we explain how to get into firmware mode. When your 3B2/300 is in multi-user mode, execute the following:

1. Login as root.
2. cd /
3. shutdown -y -i5 -g0 (this will bring the machine down to firmware mode).
4. Insert the DEVTOOLS floppy into the floppy disk drive.
5. Enter firmware password: mcp, this password can be changed. If mcp does not work, contact the system administrator for the correct passsword.
6. Enter boot.

---

* UNIX is a trademark of Bell Laboratories.

7. Enter 0, specifing the floppy disk drive as the boot device.

The system will display the following:

```
DEVTOOLS BOOT

<date of issue>
---------- CRC MODE ----------

          Floppy formatter - type formflop
            Hard formatter - type formhard
Disk to disk copy high speed - type ddhs
 Disk to disk copy by sector - type dd
            Disk <--> mem copy - type d-m
          Defect table builder - type defect
            Write sanity track - type fixdisk
  Duplicate utilities floppy - type duf
          Change boot defaults - type chgboot
                        Quit - type q

Command?
```

This will be referred to as the "DEVTOOLS menu screen" throughout the rest of this guide. Examples for each command usage will begin from this screen.

2. *FLOPPY FORMATTER*

Floppies can be formatted in firmware or multiuser mode. DEVTOOLS contains *formflop*, this will format and optionally verify floppies in the firmware mode.

2.1 FORMFLOP USAGE

To use *formflop*, enter formflop. The program will respond with the following:

```
FLOPPY DISK FORMATTER

Verify format? (n) - default is no verify pass.

Insert Floppy to be formatted, type go when ready (q to quit) . . .

Cylinders formatted
 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27
28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52
53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77
78 79
< if your answered yes to verify >
Verifying cylinders formatted
 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27
28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52
53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77
78 79
Writing defect map to floppy

DONE

<DEVTOOLS command screen>
```

2.2 FLOPPY FORMATTER ERROR MESSAGE

The floppy formatter program contains 14 error messages. Each message will be identified below and explained in the following paragraph.

1. Floppy does NOT have defect mapping capabilities!
2. FLOPPY NOT FORMATTED CORRECTLY!
3. FORMFLOP: Unable to format disk, cannot recalibrate floppy drive!
4. FORMFLOP: Unable to format disk, write protected!
5. FORMFLOP: Unable to format track = X, side 0
6. FORMFLOP: Unable to format track = X, side 1
7. write track X error
8. FORMFLOP: Unable to verify disk, cannot recalibrate floppy drive!
9. home during seek retry failed
10. track X seek failed
11. home during read retry failed
12. side X, sector X failed
13. write of defect map failed
14. write of pdsector failed

Error message with an "X" in them are replace with a number. Error message number 1 tells us that the floppy does not have error map capabilities, either from a bad format or from verifying a floppy that was formatted on a SVR1 system (you should not see this message because it is only given when verifying a previously formatted floppy). Error message number 2 tells us that the format is bad, you should get this whenever the format fails. Error messages 3 and 8 tells us that the floppy drive or floppy disk controller has a hardware problem, you should replace the floppy drive before replacing the system board. Error message 4 states that we need to remove the write protect sticker. Error messages 5 and 6 tell us what sector of the floppy could not be formatted, you may wish to try reformatting or throw the diskette away. Error message 7 tells us what track a write error has occurred on, this usually follows error message 5 or 6. Error messages 9, 10 and 11 state that an error occurred during a seek, this could be a hardware problem, usually loose cables or a bad floppy disk drive. Error message 12 states the side and sector that failed during the verification pass, try reformatting or throw the diskette away. Error message 13 states that the defect table could not be writen to the diskette, this should never be seen if the verify pass completes. Error message 14 states that the disk drive physical description sector write failed, should never be seen if the verify pass completes.

3. HARD DISK FORMATTER

Hard disks can only be formatted from firmware mode. Remember, formatting a disk will remove all user data on the disk, only the sanity track and defect maps will remain. *Formhard* automaticaly maps in the defect table builder program *(defect)* to ensure that the hard disk contains the proper defect map and the sanity track program *(fixdisk)* to write the sanity track. Formatter will format, build the defect table and install the sanity pattern on the diagnostic cylinder.

3.1 HARD DISK FORMATTER USAGE

For this example we will format hard disk number 1, it is assumed that you have already built the back-up defect table and defect map. If this is not true, go to the section on Defect Table Builder Usage before continuing.

1. To use the hard disk formatter, enter:

formhard from the DEVTOOLS screen menu.

3. The program will respond:

   Format which disk [0 or1] (0) ?

4. Since we are working with disk drive number 1, enter: 1.

5. The program will respond:

   A list of current config and defect table values will be printed.
   To keep the current value, enter <return>, to change, type the new value.

   Current config table values:
   Drive Id: 3 - ?

6. The disk drives physical ID is correct, enter: <cr>.

7. The program will respond:

   Number cylinders: 697 - ?

8. The number of cylinders is correct, enter: <cr>.

9. The program will respond:

   Number tracks/cyl: 5 - ?

10. The number of tracks per cylinder is correct, enter: <cr>.

11. The program will respond:

    Number sectors/track: 18 - ?

12. The number of sectors per track is correct, enter: <cr>.

13. The program will respond:

    Number bytes/sector: 512 - ?

14. The number of bytes per sector is correct, enter: <cr>.

15. The program will respond:

```
The following items finish describing the disk.
They correspond to the sizing information just entered.

     Logical start of disk: 90
Backup defect map location: 1
    Backup defect map size: 512
       Defect map location: 2
           Defect map size: 512
        Error log location: 89
            Error log size: 512
Number of relocation areas: 1
     Relocation area start: 3
      Relocation area size: 86

defect map read ok
back-up defect table read ok

The back-up defect table is placed on KS-spec disks by the disk
manufacturer. If this disk is a KS-spec disk, the table should be present
and left untouched by devtools. If the table is not present (Whethter
destroyed on a KS-spec disk or the disk is pre-KS) the back-up defect table
can be modified as defects are newly entered or edited below.

Modify back-up defect table? [yes or no] (no)
```

16. This time around we don't want to modify the back-up defect table, since we have already built it, enter: <cr>.

17. The program will respond:

```
If the back-up defect table is on this disk, you may choose to
force the defect map to agree with it by having devtools automatically
regenerate the defect map from the bach-up table. This will remove any
NEW defects that the formatter or bad block handling found, but will put
the defect map in the initial state.

Re-create defect map? [yes or no] (no)
```

18. Since we have already built the defect map, enter: <cr>.

19. The program will respond:

```
Current defects are:


         BACKUP          |          MAPPED
    cyl head byte length |  bad        good
-------------------------+-------------------
0:  backup empty         | 00b5020f - 00000003
1:  backup empty         | 00b50210 - 00000004

Next relocation sector is 5

Type 'new' to enter ALL defects,
     'edit' to modify current defects,
     <cr> to keep current list:
```

20. This defect map display should look like the one we built in the defect table builder usage example, excluding the backup empty messages that are given instead of the cylinder, head, byte and length counts. If there are discrepancies in the mapped display, correct the error, if not enter: <cr>.

21. The program will respond:

```
Cylinders Formatted:
 0 5 10 15 20 25 30 35 40 45 50 55 60 65 70 75 80 85 90 95 100 115 120
125 130 135 140 145 150 155 160 165 170 175 180 185 190 195 200 205 210
215 220 225 230 235 240 245 250 255 260 265 270 275 280 285 290 300 305
310 315 320 325 330 345 350 355 360 365 270 375 380 385 390 395 400 405
410 415 420 425 430 435 440 445 450 455 460 465 470 475 480 485 490 495
500 505 510 515 520 525 530 535 540 545 550 555 560 565 570 575 580 585
590 595 600 605 610 615 620 625 630 635 640 645 650 655 660 665 670 675
680 685 690 695
Cylinders Verified:
 0 5 10 15 20 25 30 35 40 45 50 55 60 65 70 75 80 85 90 95 100 115 120
125 130 135 140 145 150 155 160 165 170 175 180 185 190 195 200 205 210
215 220 225 230 235 240 245 250 255 260 265 270 275 280 285 290 300 305
310 315 320 325 330 345 350 355 360 365 270 375 380 385 390 395 400 405
410 415 420 425 430 435 440 445 450 455 460 465 470 475 480 485 490 495
500 505 510 515 520 525 530 535 540 545 550 555 560 565 570 575 580 585
590 595 600 605 610 615 620 625 630 635 640 645 650 655 660 665 670 675
680 685 690 695

Writing format information.
Writing sanity pattern
Disk 1 correctly formatted
DONE

<DEVTOOLS menu screen>
```

22. The format program will print a message for every 5 cylinders that it formats, this may be a waste of time but it does let you know it is working. The format is now complete, with the sanity track in place. You can now load the 3B2 core UNIX package or use the partition commands in sysadm, if this will be used as a second disk drive.

3.2 FORMAT HARD DISK ERROR MESSAGES

The hard disk format program has sixteen error messages associated with it. Each will be identified and then explainned in the following paragraph.

1. Invalid disk number, D!
2. FORMHARD: invalid disk info
3. FORMHARD: Could not format disk
4. FORMHARD: Could not respecify controller
5. FORMHARD: Could not verify disk
6. FORMHARD: unable to write format info
7. FORMHARD: Sanity pattern write failed
8. FORMHARD: Sanity pattern readback failed
9. FORMHARD: Seek to cylinder C failed

10. FORMHARD: Format Write of track T of cylinder C failed

11. FORMHARD: Verify Read of cylinder C head H failed due to failing sector S Add to defect maps [y, n, q(uit)] (y) ?

12. Aborting verify!!!

13. SAVE: Could not recal drive %d

14. SAVE: Could not write manufacturer defect table

15. SAVE: Could not write defect table

16. SAVE: Could not write physical info table

In the above error message: D = disk drive number, C = cylinder number, H = head number, S = sector number, and T = track number. Error number 1 is given when you enter an unknown disk number, only 0 and 1 are supported. Error message 2 is given when when a corrupted physical description file is read, make sure your physical description information is correct. Error message 3 is given when the format program gives up trying and exits. Error message 4 states the driver and disk controller are not communicating, usually this is a bad system board. Error messages 5 and 13 are printed when the verify program is tired of trying to verify and exits. Error message 6 is printed when the drive cannnot talk to the disk drive, there are no acknowledgements coming from the disk drive. Error message 7 and 8 pertain to the sanity track: first write then read it back, see the section on FDXDISK. Error message 9 is printed when a seek fails, make sure your physical description information is correct. If this error keeps occuring replace the disk. Error message 10 reports the locations of errors as they occur, you may want to write these down and enter them into the defect map (you can wait and let error message 11 correct these). Error message 11 will let use modify the defect map with the information found during the verify read-back, caution should be taken: if more than one or two errors occur the format has usually failed. Reformat before entering all these into the defect map or you will run out of defect entries! Error message 12 is printed when the verify program exits, you should reformat until the verify passes or your asking for trouble. Error message 13 states the driver and disk are having problems communicating, usually this is a disk drive problem or possibly flakey cables. Error messages 14, 15 and 16 state that the back-up defect table, defect map and physical information files could not be writen back out to the disk after the formatting was completed. This means a lot of work, you will have to rebuild the physical information file and the back-up defect table, then force the building of the defect map. When your finished with that, start the format all over again. If this happens more than once you probably have a bad disk.

4. DISK TO DISK HIGH SPEED COPY

Disk to disk copies can be completed extremely fast while in the firmware mode. The default values for this command will copy the boot block, root file system and the user file system from disk drive 0 (WREN 32MB) to disk drive 1 (WREN 32MB). This command *ddhs* assumes that both disks have defect tables and are formatted.

4.1 DISK TO DISK HIGH SPEED USAGE

We will use the default values in this example, hard disk 0 copied to hard disk 1. The default values begin on track 5 and continue until the drives own characteristic are reached (track 0, sector 5), this excludes the physical information, back-up defect table and the defect map; we do not want to write over these. The procedure is as follows:

1. From the DEVTOOLS menu screen enter: ddhs.

2. The program will respond:

HIGH SPEED HARD DISK TO HARD DISK TRACK COPY

WARNING, This routine assumes a defect table on disk.
If none exists, use defect/format to create it before proceeding.

Tracks containing defects will be transferred a sector at a time.

source disk [0 for id0, 1 for id1] (0):

3. We are using the defaults, enter: <cr>.
4. The program will respond:

   source start track (5):

5. We are using the defaults, enter: <cr>.
6. The program will respond:

   destination disk [0 for id0, 1 for id1] (1):

7. We are using the defaults, enter: <cr>.
8. The program will respond:

   destination start track (5):

9. We are using the defaults, enter: <cr>.
10. The program will respond:

    number of tracks (3480):

11. We are using the defaults, enter: <cr>.
12. The program will respond:

    sr = sector read, sw = sector write, rr = re-read track, rw = re-write track

    Tracks Transferred:
    25 50 75 100 125 150 175 200 sr-211 225 250 275 300 325 350 sr-369 375 sr-399 400 425 450 475 500 525 550 575 600 625 650 675 700 725 750 775 800 sr-812 825 850 875 900 sw-907 925 950 975 1000 1025 1050 1075 1100 sr-1455 1475 1500 1525 1550 1575 1600 1625 sr-1633 1650 1675 1700 1725 1750 1775 1800 sr-1820 1825 1850 1875 1900 1925 1950 1975 2000 2125 sr-2146 2150 2175 2200 2225 2250 2275 2300 2325 2350 2375 2400 2425 2450 2475 2500 2525 2550 2575 2600 2626 2650 2675 2700 2725 2750 2775 2800 2825 2850 2875 2900 2925 2950 2975 3000 3025 3050 3075 3100 3125 3150 3175 3200 3225 3250 3275 3300 sr-3318 3325 sr-3333 3350 sr-3373 3375 3400 3425 3450 3475
    DONE with 1 total retries

    <DEVTOOLS menu screen>

The program reports on its activity by printing every 25 sectors. Periodically it prints the sector it is currently reading (sr) or writing (sw). If it has to retry a read (rr) or write (rw) it will print the track number. When completed the program will report the total number of retries made during the disk copy.

4.2 DISK TO DISK HIGH SPEED COPY ERROR MESSAGES

The *ddhs* command has fifteen error message associated with it. Each will be identified and discussed in the following paragraph.

1. DDHS: invalid source disk number
2. DDHS: invalid destination disk number

3. Converting Destination disk from 1.0 to 1.1 format
4. DDHS: Unreadable to convert disk X
5. DDHS: Unreadable/Invalid Phys Info on Disk X
6. DDHS: Unreadable/Invalid Defect Info on Disk X
7. DDHS: Unreadable Maps on Disk X
8. DDHS: Read Track T Failed (cyl = C, head = H)
9. DDHS: Write Track T Failed (cyl = C, head = H)
10. GET_PUTTRK: Seek to S failed
11. GET_PUTTRK: DMA setup failure
12. Failed all retries
13. CONV_DISK: read old sector 0 failed
14. CONV_DISK: write new sector 0 failed
15. CONV_DISK: write of new defect map failed

In the above error messages: X = disk drive number, C = cylinder, H = head, S = sector, and T = track. Error messages 1 and 2 report that an invalid source or destination disk number has been specified, only 0 and 1 are supported. Error message 3 states that the destination disk drive is being converted from 1.0 (SVR1) format to 1.1 (SVR2) format. Error message 4 is given when the conversion fails, usually the source disk can not be read. Error messages 5, 6 and 7 let you know that the physical information, and defect tables can not be read or they have been corrupted. Use the defect program to fix these problems. Error messages 8 and 9 report read or write failures, giving the location of the error. Error message 10 states that a seek error has occurred, giving the sector number that has failed. Error message 11 states the DMA setup has failed, this usually means you have a bad system board. Error message 12 states that every retry has failed, the program will usually terminate when you get this. Error messages 13 and 14 report that a sector 0 read or write has failed, make sure the disk contains the correct physical information with the defect program. Error message 15 states the new defect map write has failed. You may want to use the format program to update the disk to 1.1 (SVR2) manually, then do the copy.

5. DISK TO DISK COPY BY SECTOR

Disk to disk copies by sector are possible with the *dd* command. This utility is used for disk transfers between hard disks or floppy disk, you specify source and destination locations. This command assumes that both disks have a defect table and are formatted.

5.1 DISK TO DISK COPY BY SECTOR USAGE

Lets walk through an example, we will copy sectors 200 thru 1000 from integral hard disk 0 to integral hard disk 1.

1. From the DEVTOOLS menu screen, enter: dd.
2. The program will respond:

   WARNING, This routine assumes a defect table on disk.
   If none exists, use defect/format to create it before proceeding

   source disk [id or if] (id) :

3. The first message warns you about an unformatted disk drive, make sure both disks have already been formatted before continuing. Since we are copying from intergal hard disk 0, just enter: <cr>.
4. The program will respond:

which [0 or 1] (0) :

5. We want disk 0 as our source, enter: <cr>.
6. The program will respond:

   source start block:

7. We want to begin copying at block 200, enter: 200.
8. The program will respond:

   destination disk [id or if] (id) :

9. We want intergral hard disk 1, enter: <cr>.
10. The program will respond:

    which [0 or1] (0) :

11. Since we want disk 1, enter: 1.
12. The program will respond:

    destination start block:

13. We will put the blocks in the same location on the second disk drive, enter: 200.
14. The program will respond:

    number of sectors:

15. We want to copy 200 thru 1000 to the second disk, enter: 800.
16. The program will respond:

    Sectors Transferred:
    200 218 236 254 272 290 308 328 344 362 380 398 416 434 452
    470 488 506 524 542 560 478 596 614 632 650 668 686 704 722
    740 758 776 794 812 830 848 866 884 902 920 938 956 974

    DONE

    <DEVTOOLS menu screen>

The program will print a message to you for every 18 sectors copied to the destination location. When complete, DONE will be printed and you will be returned to the DEVTOOLS menu screen.

5.2 DISK TO DISK COPY BY SECTOR ERROR MESSAGES

The *dd* command has five error messsage associated with it. Each will be identified and desrcibed in the following paragraph.

1. DD: Floppy Defect Map Check Failed
2. DD: Invalid Hard Disk Number
3. DD: Invalid Disk
4. D-M or DD: Floppy Defect Map Invalidation Failed
5. DACS: R/W sector S fail

Error message 1 reports the floppy's defect map is not in place, format the floppy. Error message 2 is printed if you enter an invalid hard disk number, 0 and 1 are support. Error message 3 is printed when an invalid disk type is entered, only id (integral hard disk) and if (integral floppy disk) are supported. Error message 4 is printed when the floppy diskette defect map could not be invalidated, basically the write to the floopy diskette

failed. Error message 5 reports that a read or write to the disk has failed (R = read, W = write and S = sector number). If it is a read failure make sure the disk is formatted. If it is a write failure make sure the diskette is not write protected, in the case of a hard disk write failure make sure it is formatted properly, you may want to run the format program: it will report read and write errors along with the actual address.

6. DISK TO MEMORY COPY

The *d-m* command allows you to copy data from floppy disk drive or one of the hard disks (0 or 1) to main memory. Presently you may not copy from memory to disk. The only draw back is: you must copy by sectors, 512 byte chuncks.

6.1 DISK TO MEMORY COPY USAGE

To use *d-m*, use the following steps:

1. From the DEVTOOLS menu screen, enter: d-m.
2. The program will respond with the following:

   Disk to Memory Copy
   WARNING, This routine expects maps to be in place on disk
   If they are not, use defect/format to build them

   read, write, or quit? [r,w,q] (r) :

3. If your disk doesn't have defect maps and has not been formatted, do formhard before continuing. For this example we will read the first sector of hard disk drive 0 into memory. The prompt: read, write, or quit defaults to read (r). Enter: <cr>.
4. The program will respond with:

   disk [if or id] (if) :

5. The default is the integral floppy disk, since we are reading the first sector of the integral hard disk 0, enter: id.
6. The program will respond with the following:

   which [0 or1] (0) :

7. The default is integral hard disk 0, enter: <cr>.
8. The program will respond with:

   Enter the starting block (0) :

9. The default is the beginning of the device, enter: <cr>.
10. The program will respond with:

11. Enter the memory address: 0x

    This asks for the beginning address in main memory, hexidecimal physical location, we will start at 0x200400.

12. The program will respond with:

    Enter the number of blocks:

13. Since we are only copying one block, enter: 1.
14. The program will respond with:

```
Sectors Copied:
 0
DONE

<DEVTOOLS menu screen>
```

Sectors copied are printed on the screen, followed by the number of the sector copied to main memory, if we were copying 100 sectors it would count to 100.

6.2 DISK TO MEMORY COPY ERROR MESSAGES

There are six error messages associated with the disk to memory copy command. Each will be identified below and explained in the following paragraph.

1. Invalid Disk!
2. D-M: Floppy Defect Map Invalidate Failed
3. D-M: Floppy Defect Map Check Failed
4. D-M: Unreadable/Invalid Phys Info on Disk X
5. D-M: Unreadable Maps on Disk X
6. idX CRC error at disk address 0xY (Z retries)

Error message 1 is given when an invalid disk identification number was entered. The only valid entries are 0 or 1. Error messages 2 and 3 pertain only to the integral floppy disk drive. The driver doesn't know how the diskette is partitioned, use a different floppy. Error message 4 states that physical description table for disk drive X could not be read, or contained invalid information, use the defect program to repair the physical description table. Error message 5 states the defect maps on disk drive X could not be read, use the defect program to rebuild the defect tables. Error message 6 states disk drive X has had a CRC error at disk address 0xY and has retied Z amount of times to correct it. This usually means that the disk drive has gone bad, you can try to map this location out and reformat before replacing the disk drive.

7. DEFECT TABLE BUILDER

The defect table builder, ***defect***, is invaluable for building defect maps and reading the factory defect map. Presently this is the only tool for building defect maps on 3B2/300 supported disk drives, ST-506 interface. A special note about this command: the editor has some flaws, sometimes you get into edit mode and you can't get out. You can save yourself a lot of time by using the new command and typing carefully.

7.1 DISK MAP BUILDER USAGE

The defect map builder program installs data that must match the volume table of contents (VTOC) description files located in /etc/vtoc. Each type of disk drive has a specific identification number associated with it. Presently, AT&T supports three disk drives, each described in the following table.

Manufacture	Model	Size	ID	Cylinders	Tracks/Cylinder	Sectors/Track	Bytes/Sector
CDC	Wren	30M	3	697	5	18	512
CDC	Wren II	72M	5	925	9	18	512
Seagate		10M	2	306	4	18	512

Use the information from this table to answer question the defect program asks. Lets take an example walk through using a CDC Wren 30M disk drive.

1. From the DEVTOOLS menu screen, enter: defect.
2. The defect program will respond:

Building defect table ONLY (no formatting)

Which disk [0 or 1] (0)

3. Reply with the disk drive number you wish to display or modify the defect table on. In our case we will use drive number 1, enter: 1.

4. The program will respond with:

A list of current config and defect table values will be printed.
To keep the current value, enter <return>, to change, type the new value.

Current config table values:
Drive Id: 2 - ?

5. Enter the drive id number if different than the number indicated, in this case we need to enter: 3.

6. The program will respond:

Number cylinders: 306 - ?

7. Enter the number of cylinders for your disk drive if different than display value, in this case we enter: 697.

8. The program will respond:

Number tracks/cyl: 4 - ?

9. Enter the number of tracks per cylinder if different than displayed value, in this case enter: 5.

10. The program will respond:

Number sectors/track: 18 - ?

11. Enter the number of sectors per track if different than displayed, in this case enter: <cr>.

12. The program will respond:

Number bytes/sector: 512 - ?

13. Enter the number of bytes per sector if different than displayed value, in this case enter: <cr>.

14. The program will respond:

The following items finish describing the disk.
They correspond to the sizing information just entered.

```
    Logical start of disk: 90
Backup defect map location: 1
   Backup defect map size: 512
     Defect map location: 2
        Defect map size: 512
      Error log location: 89
         Error log size: 512
Number of relocation areas: 1
    Relocation area start: 3
     Relocation area size: 86

defect map read ok
back-up defect table read ok
```

The back-up defect table is placed on KS-spec disks by the disk manufacturer. If this disk is a KS-spec disk, the table should be present and left untouched by devtools. If the table is not present (Whethter destroyed on a KS-spec disk or the disk is pre-KS) the back-up defect table can be modified as defects are newly entered or edited below.

Modify back-up defect table? [yes or no] (no)

15. In this example we will construct the back-up defect table, just for the experience of working with a disk that has been completely wiped out. Enter: yes.

16. The program will respond:

Current defects are:

```
     BACKUP          |      MAPPED
cyl head  byte length |  bad        good
----------------------+-----------------
```

Next relocation sector is 3

Type 'new' to enter ALL defects,
'edit' to modify current defects,
<cr> to keep current list:

17. In this example the map is presently empty. For this reason we need to enter all the defects the disk drive has, enter: new.

18. The program will respond:

Enter type of defect byte count, [bc or bc/48] (bc) ?

19. This is asking for the way you wish to enter defect locations: byte count or byte count/48. The defect list provided by the manufacturer, usually typed on a stick located on the disk drive, is in byte count values. Enter carriage return for bc mode.

20. The program will respond:

```
Enter defective sectors one per line, as cylinder head bytecount length.
Defect length defaults to 1 if not entered!
End with 'q', restart with 'new'
bad sector = ?
```

21. This is telling you how to enter the defects location:

    1. Cylinder number.
    2. Head number.
    3. Bytecount, length in bytes from the begining of sector.
    4. Length of defective area, defaults to 1 if no value is entered.

    Lets enter the defects for the drive used in this example, program prompts and defect information will be together.

22. Interactive display:

```
bad sector = ? 181 2 9114
Bad sector 00b5020f maps to good sector 00000003 (defect 1)
Bad sector 00b50210 maps to good sector 00000004 (defect 2)
bad sector = ? q

Current defects are:

          BACKUP          |          MAPPED
  cyl head byte length |  bad        good
-----------------------|-----------------
 0: 181    2 9114      1 | 00b5020f - 00000003
 1:                      | 00b50210 - 00000004

Next relocation sector is 5

Type 'new' to enter ALL defects,
    'edit' to modify current defects,
    <cr> to keep current list:
```

23. This drive only has one defect, but it happens to be on a border so two locations are mapped out. Lets keep this map and write it to disk, enter: <cr>.

24. The program will respond:

```
Writing format information.
<DEVTOOLS menu screen>
```

25. Now we have built the back-up defect table or manufacturer defect table, which ever you would like to call it. The next step is to construct the defect table that will be used to format the disk drive. Enter: defect.

26. The program will respond:

```
Building defect table ONLY (no formatting)

Which disk [0 or 1] (0)
```

27. We are working with drive number 1, enter: 1.

28. The program will respond with:

A list of current config and defect table values will be printed.
To keep the current value, enter <return>, to change, type the new value.

Current config table values:
Drive Id: 3 - ?

29. Note that the disk drives physical ID is now correct, enter: <cr>.

30. The program will respond:

Number cylinders: 697 - ?

31. Note that the number of cylinders is now correct, enter: <cr>.

32. The program will respond:

Number tracks/cyl: 5 - ?

33. Note that the number of tracks per cylinder is now correct, enter: <cr>.

34. The program will respond:

Number sectors/track: 18 - ?

35. Note that the number of sectors per track is now correct, enter: <cr>.

36. The program will respond:

Number bytes/sector: 512 - ?

37. Note that the number of bytes per sector is now correct, enter: <cr>.

38. The program will respond:

The following items finish describing the disk.
They correspond to the sizing information just entered.

```
    Logical start of disk: 90
Backup defect map location: 1
   Backup defect map size: 512
     Defect map location: 2
        Defect map size: 512
      Error log location: 89
         Error log size: 512
Number of relocation areas: 1
    Relocation area start: 3
     Relocation area size: 86
```

defect map read ok
back-up defect table read ok

The back-up defect table is placed on KS-spec disks by the disk manufacturer. If this disk is a KS-spec disk, the table should be present and left untouched by devtools. If the table is not present (Whethter destroyed on a KS-spec disk or the disk is pre-KS) the back-up defect table can be modified as defects are newly entered or edited below.

Modify back-up defect table? [yes or no] (no)

39. This time around we don't want to modify the back-up defect table, since we had already built it, enter: <cr>.

40. The program will respond:

    If the back-up defect table is on this disk, you may choose to
    force the defect map to agree with it by having devtools automatically
    regenerate the defect map from the bach-up table. This will remove any
    NEW defects that the formatter or bad block handling found, but will put
    the defect map in the initial state.

    Re-create defect map? [yes or no] (no)

41. Since we have modified (built) a new back-up defect table, force the rebuilding of the defect map. Enter: yes.

42. The program will respond:

    Automatically re-constructing defect map from backup map
    Bad sector 00b5020f maps to good sector 00000003 (defect 1)
    Bad sector 00b50210 maps to good sector 00000004 (defect 2)

    Current defects are:

```
          BACKUP          |          MAPPED
  cyl head byte length |  bad        good
-----------------------|-----------------
0: 181   2 9114      1 | 00b5020f - 00000003
1:                     | 00b50210 - 00000004
```

    Next relocation sector is 5

    Type 'new' to enter ALL defects,
        'edit' to modify current defects,
        <cr> to keep current list:

43. The automatic re-construction has built a new defect map identical to the present back-up map. If you would have answered no, the old defect map would have been displayed. If the drive is sane enough to read the old defect map read it! You can find defects that are not listed on the manufacturer's defect list, if new defects have been discovered. If these error are not correct go back through the original steps to correct them, otherwise enter: <cr>.

44. The program will respond with:

    Next relocation sector is 5

    Writing format information.
    <DEVTOOLS menu screen>

45. This tells us that the next relocation sector is sector 5, this would be the location used to map out the next defect. The information needed to format the disk is now in place, use *formhard* to finish constructing a usable disk drive.

7.2 DEFECT MAP BUILDER ERROR MESSAGES

The defect map builder has thirteen error messages associated with it. Each will be identified below and discussed in the next paragraph.

1. Invalid disk number

2. FORMHARD: Could not read physical info on disk %d
3. FORMHARD: Invalid physical info on disk %d
4. Assuming default values for physical info
5. defect map unreadable
6. WARNING: Manual reconstructing of defect map REQUIRED
7. back-up defect table unreadable
8. D: backup empty | XXXXXXXX - XXXXXXXX
9. D: CYL HEAD BYTE LENGTH | map empty
10. D: no backup location | XXXXXXXX - XXXXXXXX
11. D: | second sector missing in defect map
12. D: CYL HEAD BYTE LENGTH | no mapped sector!
13. D: incorrectly mapped | XXXXXXXX - XXXXXXXX

Error number 1 states that the disk number you entered is invalid, only 0 or 1 is supported. Error message 2 states that the driver could not read the physical description file on disk drive X, if you are repairing the disk ignore this the first time you get it. Continue through the defect entry procedure and the message should go away the next time you enter the defect program. If the error doesn't go away the second time you may need to replace the disk. Error message 3 states the physical information for this disk is invalid, some part of the description is invalid, ie 1000 tracks per cylinder. Go through the defect procedure and rebuild the physical information table, this should go away the next time through. If not replace the disk. Error message 4 states the defect map is unreadable, go through the defect mapping procedure, this should go away the second time through, if not you can try formatting the drive and then build the defect table (you must go back and format the disk again, since the defect map was empty during the first format). Error message 5 states the defect map has been corrupted and must be rebuilt, the back-up defect table may still be OK. Check the back-up table and force a rebuild of the defect map if it is OK. If the back-up defect table has been corrupted you will have to rebuild it before forcing the automatic reconstruction of the defect map. Error message 7 states the back-up defect table is unreadable, you will have to rebuild it. Error messages 8 thru 13 are given during the defect table display routines, usually these are just telling you the difference between the back-up defect table and the defect map (D = defect number and X = disk address). You may wish to rebuild the back-up defect table to show all of the defect areas, since the hard disk error logger and fixer will only update the defect map. You should always clean up the table when you receive error numbers 9, 11, 12 and 13.

8. WRITING SANITY TRACK

The *fixdisk* command will rewrite the sanity track on hard disk 0 or 1.

8.1 FIXDISK USAGE

In this example we will fix the sanity track on integral hard disk 0.

1. To use *fixdisk*, enter: *fixdisk*. The program will respond with the following:

   Which disk? [0 or 1] (0)

2. We want to fix drive 0, enter: <cr>.
3. The program will respond:

FIXING SANITY TRACK ON DISK X

Sanity Pattern is in Place on Disk X
DONE

<DEVTOOLS command screen>

In the above statements, X = disk identification number. If you reply with a carrage return, default values will be used. The sanity track is now in place.

8.2 WRITING SANITY TRACK ERROR MESSAGES

Fixdisk has four possible error messages:

1. FIXDISK: Could not respecify controller
2. FIXDISK: Sanity pattern readback failed
3. FIXDISK: Sanity pattern write failed
4. 1-02 DISK SANITY FAILURE

Error message 1 states the driver is having a problem with the disk controller, this may be a system board problem. Error message 2 states the driver couldn't read the sanity track after it wrote it, try to rewrite the sanity track: if this fails replace the disk drive. Error message 3 states the sanity track could not be writen to the diagnostic cylinder, try to format the disk before trying again: the format may have been corrupted. Error message 4 is given by the firmware diagnostics, try to install a sanity pattern before replacing the disk drive.

9. DUPLICATING UTILITY

Duf is used to duplicate the DEVTOOLS utility program, on a pre-formatted floppy.

9.1 DUPLICATING UTILITY USAGE

This is a fast and easy way to make a copy of DEVTOOLS. Follow the direction below.

1. From the DEVTOOLS menu screen, enter: *duf.*
2. The program will respond with the following:

   Duplicate utilities floppy

   assumes blank formatted floppy available for drive . . .

   insert floppy and type 'go', if none available, type 'q'

   At this time put in your blank formatted floppy and enter: go.

3. The program will respond with the following:

   Boot block copied to floopy disk
   Now copying 76 sectors of program
   1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25
   26 27 28 292 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47
   48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69
   70 71 72 73 74 75 76

   <DEVTOOLS command screen>

4. Note: 76 maybe replaced by any number, depending on the version of DEVTOOLS you have. This completes the copy.

9.2 DUPLICATING UTILITIES FLOPPY ERROR MESSAGES

There are two error messages associated with the duplication utility.

1. DUF: write failure on sector 0
2. DUF: write failure on sector N

When you get the failure on sector 0, this usually means that the floppy is write protected. N denotes the sector an error has taken place.

10. CHANGE BOOT DEFAULTS

Chgboot is used to change the default boot device, this is stored in the NVRAM.

10.1 CHANGE BOOT DEFAULT USAGE

The *chgboot* default values will setup integal floppy disk 0 to be the booting device, unix to be the boot file and disk section 1 to be the save device. We will use values to make the integral hard disk 0 to be the auto boot device in this example.

1. From the DEVTOOLS menu screen, enter: *chgboot*.
2. The program will respond with the following:

   new boot device slot number [0-12] (0):
3. This is asking for the slot number of the boot device, this should always be 0 (system board), enter: <cr>.
4. The program will respond with:

   new boot device subdevice number [0 - 15] (0):
5. This is asking for the subdevice number, 0 = integral floppy disk drive, 1 = integral hard disk drive 0 and the other numbers are used as growth devices are added, ie second hard disk. For the integral hard disk 0 enter: 1.
6. The program will respond:

   save device: 01 ?
7. This is asking for the swap device, or the place to dump the kernel in case of a panic. Use the default value unless you have modified the file system structures. Enter: <cr>.
8. The program will respond:

   new boot file name (unix):
9. This is asking for the name of the file to execute, you should leave this as unix, enter: <cr>.
10. The program will respond:

    save name: unix ?
11. This is asking for the name to put on the save file, use the same name as used for the boot file name, enter: <cr>.
12. DEVTOOLS menu screen.

You have now modified the boot defaults. If these defaults are wrong or the name of the boot file can not be found the machine will go into firmware mode during the power-up sequence and display:

SYSTEM FAILURE: CONSULT YOUR SYSTEM ADMINISTRATION UTILITIES GUIDE

10.2 CHANGE BOOT ERROR MESSAGES

Chgboot has one error message:

1. CHGBOOT: device write failed

When you recieve this message, NVRAM can not be wrote. Consult your service representative.

11. CLOSING COMMENTS

Although DEVTOOLS has some bugs, it is invaluable to someone working with the 3B2. This maual should help you thru most of your problems, in the future this may become a supported product if the demand is high enough. Good luck and have fun, but be carefull if your working on a drive that has valuable data.

CU-T2CU132130-MHM-mm **M. H. Murnighan**

Copy to

H. L. Mitchell - Computer Systems Center
All members T2CU132100 - Computer Systems Center

NOTE 2 cont.

2. The formhard command will prompt the user for configuration values of the disk, this information is dependent upon the type of disk that is equipped with the system. The following configuration table values should be used with the supported drives.

Seagate = 10meg.

Wren = 32meg.

```
            Drive Id: 2  - ?  use 2 for Seagate, 3 for Wren
     Number cylinders: 304- ?  304 for Seagate, 697 for Wren
    Number tracks/cyl: 4  - ?  4 for Seagate, 5 for Wren
 Number sectors/track: 18 - ?  same for both drives
 Number bytes/secto : 512- ?  same for both drives
```

3. The disk defect map will be printed next if it is readable from the disk. In the event that the defect map is not readable, you you will have to load it in. The defect map should be taped to the cover of the hard disk.

4. After you are finished, type q to quit. The utility will now format and verify the hard disk. If any errors occur during the format, (i.e.....R's come up next to numbers etc.), the disk is physically bad. Get another (I know, easier said than done).

5. The program will then tell you " format complete ". It will return you to firmware mode. You can now load core unix.

AT&T Information Systems

subject: 3B2 Kernal Reconfiguration date: May 4, 1984

from: F. L. Butler
LZ 02217
3L-319 x3190

ENGINEER'S NOTES

There are several system tunable parameters that can be modified to customize the operating system to ones particular needs, one particular one that need to be considered when loading application software is the 'maxmem' parameter. This identifier tells the system the maximum number of pages of memory to allow each process to consume. The default amount (or factory setting) is 128 pages or 256K of memory per process, this is not enough memory for many of our software packages. The problem can be rectified very easily in the following manner:

A. Determine the amount of memory the application you have will need. This can be found by looking at the size of the executable file, decide how many Kilobytes the file is and divide that number by 2K to determine what the maxmem parameter need to be.

B. Edit the file /usr/src/uts/3b2/cf/df and change the maxmem parameter to the value you calculated in the previous step (Assuming you have loaded the System Reconfiguration Utility Package).

C. Execute the following command at the system console in the /usr/src/uts/3b2/cf directory:

   - /etc/reconfig make df #This takes about 12 minutes.

D. Bring system down into single user mode by the following:

   - shutdown -is -g0 -y #Single User Mode command.

E. Mount the /usr file system by doing the following:

- mount /dev/idsk02 /usr #Mounting the /usr file system.

F. The last step in rebuilding the kernal is to execute the following:

- /etc/reconfig reboot #Rebooting the system with the new KERNAL.

Once system is rebooted the application should run successfully on the newly rebuilt system.

LZ-02217-ELB-flb F. L. Butler

DEV TOOLS INFORMATION

When loading in the disk defect map on a wren (32meg.) hard disk please note the following.

If the map taped to the cover of the hard disk has the statement Byte count/48 the w option is used, the format is
w cylinder # head # sector # else
The more common option is the (A) absolute.
The format for this option is
Acylinder # head # sector #

NOTE: no space here

Coutesy of : Mark Bauerlein

PROCEDURE : Entering FIRMWARE MODE after SYSTEM FAILURE message is displayed.

Firmware mode will only recognize certain terminal settings.
They are:

1) send parity = space or none
2) check parity = no
3) duplex = full
4) return key = carriage return only
5) new line = off or no
6) baud = 9600

After you have checked these settings, you can now type mcp.

Linda

ps : for those of you who were smart enough to already figure this out - please ignore. For me, it was a great revelation. Especially as far as other terminals go (not 5410 or 5420).

C. SELECTING LOCATION:

In selecting location for the 382 computer, the following factors must be considered:

1) size : 22' wide, 17' deep, 3 1/2 high.
2) weight : 30 pounds.
3) 110 AC, 15 AMP.
4) Normal office envirenment (avoid excess heat, humidity and dust.)
5) Allow for free air flow in the back of the 382 computer.
6) 382 computer can support up to 60 lbs. You can put the console terminal right on top of the 382 computer.
7) Maximum recommeded distance from a terminal to the computer is 50 feet.
8) Take printer and other peripherals into account.
9) Consider space for manuals, diskettes, and work areas.

After considering all the above factors and having arrived at a definite location to place the 382 computer, the next step is to set it up.

D. SET-UP:

. The general basic requirement for the terminal set-up is as followed:

Speed	9600
Duplex	full
8 bit ASCII interface	
No parity	

1) 4410 terminal:

Speed	9600
Return key	cr
Parity	none
Receivd line feed	NL
Duplex	Full
Screen	80
Autowrap	ON
Transparent	off
Keyclick	either(on or off)

2) 4415 terminal:

Speed	9600
Duplex	Full
Send parity	~
check parity	no
132 columns	off

Memory access	scroll
Clock	asynch
Return key	cr
Newline on LF	no
Autowrap	on
Cursor	(blink or steady)
keyclick	(on or off)
Margin bell	(on or off)
keyboard model	5420
Transmission	char
Line send	keyed
Block send	unprot
Send from	cursor
Send Edit SEQ	no
Send Graphics	no

* The serial connector that plug into the terminal side must have a 6 to 20 strap in the modular connector.

3) 4420 terminal

201	F		213	B	228	A
202	A		214	B	229	C
203	B		215	B	230	A
204	B		216	B	231	A
205	A		217-220	UNUSED	232	B
206	B				233	B
**207	A	**	221	A	234	B
208	K		222	A	235	A
209	B		223	A	236	B
210	K		224	B	237	A
211	K		225	B	238	B
212	A		226	A	239	UNUSED

* Must strap pin 6 to pin 20 in the modular connector.

** Must change 207 option from A to D when going to firmware!!!!!

4) Digital VT-100

- Hit set-up key .
- Hit the A/B key to go to set-up B
- Four windows on the bottom of the screen are highlighted.
- Move the cursor with the tab key or arrow key to position the cursor over window 4 the first box.

1 0 0 0 1 2 0 0 1 0 3 0 1 0 0 4 0 0 1 0

. Hit toggle 1/0 key to chang the 0 to 1.

1 0 0 0 1 2 0 0 1 0 3 0 1 0 0 4 1 0 1 0

. change transmit and receive speed to 9600 baud by hitting the transmit and receive key till 9600 is displayed.

5) kaypro-plus

. Use KMDM795 modem program.
. TIM to set the baud rate.
. 4 for 9600
. T for terminal mode.

the 3B2 computer requires the terminal to run at 9600 baud when it comes up. Once in unix, you can step terminal speed down with stty command (stty 1200). Kaypro-plus does not work too well at 9600 baud.
Reason for using KMDM is to capture output on screen into a CPM file for future reference or use (have a hard copy to look at).
. Ctr-1 E return to modem mode.
. T unix1 enter terminal with a CPM file created.
. Ctrl- Y open buffer to capture text.
also put ":" to let you know buffer is open.
. After capturing all the text you want,
Ctrl-R closes the buffer.
. WRT command makes the kaypro writes its buffer to disk.
. Now you have a unix1 CPM file to print out so you can look at.

6) TRS-80 model 100:

. term stat 87ile

. term stat 88n1d

	NORMAL MODE	FIRMWARE MODE
Speed 9600	8	8
8 or 7 bit	7	8
Parity	i	n
Stop bit	1	1
	e	d

* Ask Eric about this option.

BOOTING FROM FLOPPY DRIVE

Use this procedure in the event that nothing can be run or loaded from firmware mode.You must have a good idea that the hard disk is not physically damaged,and that DEV TOOLS can remedy it.

-Have the tech powerdown the 3B2. At this point, probably the only available method would be to hit the power switch.

-Remove the cover, and disconnect the two ribbon cables on the hard disk.

-Powerup the machine, and enter firmware ("SYSTEM ERROR" message will occur).

-Insert DEV TOOLS into the slot.

-Execute "boot" from firmware (from the floppy - "if").

-Replace the two ribbon cables.

-Run "DEFECT" from DEV TOOLS to rewrite the hard disk's defect map.

-Run "FIXDISK" if needed to rebuild the sanity track.If it is not evidently needed, it can't hurt to run it just in case.

-If all works without problems, powerdown the machine, replace the cover, and power back up.

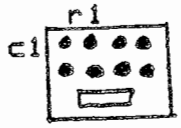
-Rewrite boot, init, root blocks as needed; and as per previous instructions.

Use this procedure as a last resource only !!

Courtesy of: Jim Lanahan

PROPER PIN CONFIGURATION FOR CONNECTOR USING MODEM WITH 382 COMPUTER

MODULAR PART OF CONNECTOR



RS232

PIN #

r1c1	4
r1c2	8
r1c3	20
r1c4	5
r2c1	7
r2c2	3
r2c3	2
r2c4	1

PIN ASSIGNMENTS FOR PARALLEL PRINTERS

PIN	SIGNAL
1	DATA STROBE
2 - 9	DATA BITS 1 - 8
10	NO CONNECTION
11	BUSY
12	PRINTER ERROR
13	SELECT
14	+- 0 VOLTS
15	NO CONNECTION
16	GROUND
17	FRAME GROUND
18	NO CONNECTION
19 - 29	GROUND
30	GROUND
31	INPUT PRIME
32	FAULT
33	GROUND
34 - 36	NO CONNECTION