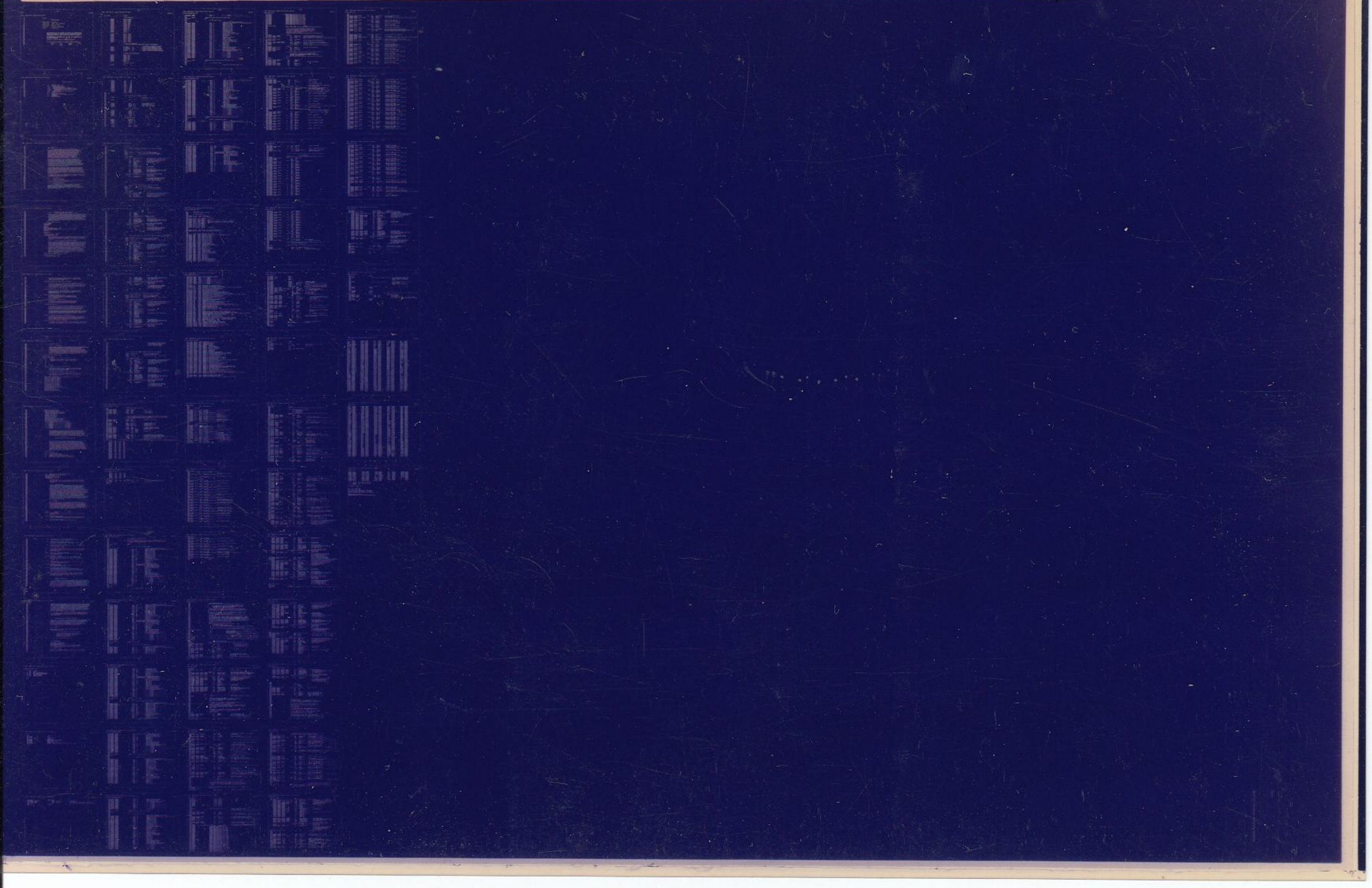
RD51/52/53 RQDX 3 Formatter AH-U110A-MC
RQDX 3 CZRQCA0 1 of 1 Juli 85



```
b 9j w
A .i
1                                                                    SEQ 000

.MAIN.  MACRO V05.02  Tuesday 23-Apr-85 14:52

                .REM %(

                                        IDENTIFICATION
                                        --------------

                PRODUCT CODE:   AC-U109A-MC

                PRODUCT NAME:   CZRQCA0  RQDX3 FORMATTER

                PRODUCT DATE:   APRIL 24,  1985

                MAINTAINER:     DIAGNOSTIC ENGINEERING

                AUTHOR:         Richard Dietz


                THE INFORMATION IN THIS DOCUMENT IS  SUBJECT  TO  CHANGE WITHOUT
                NOTICE  AND  SHOULD  NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL
                EQUIPMENT CORPORATION.  DIGITAL EQUIPMENT CORPORATION ASSUMES NO
                RESPONSIBILITY FOR ANY ERRORS THAT MAY APPEAR IN THIS DOCUMENT.

                NO RESPONSIBILITY IS ASSUMED  FOR  THE  USE  OR  RELIABILITY  OF
                SOFTWARE ON EQUIPMENT THAT IS NOT  SUPPLIED  BY  DIGITAL  OR ITS
                AFFILIATED COMPANIES.

                COPYRIGHT (C) 1985 BY DIGITAL EQUIPMENT CORPORATION


                THE FOLLOWING ARE TRADEMARKS OF DIGITAL EQUIPMENT CORPORATION:

                        DIGITAL         PDP             UNIBUS          MASSBUS
                        DEC             DECUS           DECTAPE
```

TABLE OF CONTENTS

```
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
```

1.0 ABSTRACT

This formatter was written to format Winchester drives attached to a RQDX3 disk controller. All new drives being attached to the RQDX3 disk controller must be formatted so that the drive can be brought online for use by a MSCP server or in simpler terms be used by an operating system. This disk formatter is similar to the RQDX1/2 disk formatter in that the same standard DUP dialog is used and similar standard formatter questions are passed by the controller to the host user. The formatter is different from the RQDX1/2 disk formatter because a table of formatting parameters is passed to the controller. The RQDX1/2 disk controller already has these tables in its firmware.

The UIT, Unit Information Table is picked by the host user formatting his Winchester on a RQDX3 according to the drive name. Example RD51, RD52, RD53. The program then passes the UIT to the controller and the table is written on the disk. Every time the drive is brought online after being formatted the UIT is read in by the disk controller from the drive. As long as the UIT still exists on the drive it does not have to be passed in by the host user. Only if the user requests to "Down line load" information to the controller will the host user have to pick a UIT table to pass to the controller.

The UIT table contains information about the drive such as size, number of tracks per surface, etc. This information is already know for certain DEC acquired Winchester drives. These tables are usually different for the different drives manufactured. If a new or unlisted drive is to be formatted, the UIT table can be built by answering about twenty questions. Simple choose the UIT # that refers to the drive "other". This will go through all the questions that make up the UIT.

All though not a goal of the diagnostic this program can be used to run standard DUP dialog local programs such as "DIRECT". These local programs are stored in the firmware.

2.0 HOW TO RUN IT?

2.1 HARDWARE REQUIREMENTS

A RQDX3 disk controller and one or more Winchester drives configured into a Q-bus PDP-11 system.

2.2 SOFTWARE REQUIREMENTS

This diagnostic was written using DRS the Diagnostic Supervisor. The diagnostic is expected to be run under XXDP diagnostic operating system. The diagnostic uses a lot of manual intervention, answering DUP format questions send by the RQDX3 firmware. For this reason the diagnostic is APT loadable but not APT controllable.

2.3 QUESTIONS ASKED AND THEIR ANSWERS

2.3.1 HARDWARE QUESTIONS FROM DIAGNOSTIC SOFTWARE

The diagnostic is a standard DRS program with the standard DRS commands. Below I have a script of the questions asked an the answers to the initial DRS questions. The Default value for the IP address is 172150. This is standard configuration address for the first MSCP controller on a system. Any other MSCP controllers on the system will have to be in the floating address space of the IO page. The default vector address is 154 any other value between 0 777 could be used but is not suggested. If you want the default answers then just hit the "return" key on the keyboard.

Typical Diagnostic Script:

```
boot up XXDP
.RUN ZRQC??
ZRQCA0.BIN

DRSXM-A0
ZRQC-A-0
RQDX3 Disk Format Utility
Unit is RD51,RD52,RD53,or RQDX3 Proto-type Winchester drive
Restart Address is 141656
DR>START

Change HW ?  Y
# Units ?  1

IP Address 172150 ? <rtn>
Vector address 154 ? <rtn>
```

After these questions have been answered more questions will be asked as long as manual intervention is allowed on the system. If no manual intervention is allowed the diagnostic will return without formatting the drive.

2.3.2 MANUAL QUESTIONS FROM CONTROLLER FIRMWARE

Manual Questions are asked from inside the diagnostic and are not part of the P table as described in the DRS programmers guide. The first question and the UIT table questions are asked by the host pr gram all other questions are asked by the RQDX3's firmware. For purposes of international support these questions given by the controlier are not used but a message number return along with the question is used to look up the translated question contained in this diagnostic. If the message number is unknown the ASCII data is printed out as is in English. To turn off controller reported messages just set the IXE flag in the diagnostic monitor. Below is a script of the manual questions asked. Depending on how certain questions are answered will depend on what questions will be asked.

Text printed, Questions asked ,and replies:

```
MSCP Controllre model # : 019
    Microcode version # : 001
```

Every MSCP controller has a model number. The RQDX3s model number is 19. The RQDX1 model number is 7. This also reports the microcode revision number.

What local program do you want to run (A) FORMAT ?

This question asks what controller local program you want to run. Usually if not always we will want to run FORMAT. If you get curious you can write DIRECT which is a controller local program which list all the controller local programs. The default is to run the local program FORMAT. At the prompt just hit "return".

Enter date <MM-DD-YYYY>: (A) ? current date

There is no default to the date question. You must use the appropriate form to answer the date. If not the question will be asked again until it is in the correct form.
EXAMPLE 12-12-1985

Enter unit number to format <1>: (A) ?

The default unit number is unit or physical drive zero. If the drive you want to format is other then drive 0 then make sure you type the number followed by a carr age return.

Use existing Bad Block Information <N> ?

The default is no which is probable the best choice. To use the existing information would only clean the drive up and not possible correct all the problems.

Use Down Line Load <N> ? Y

If this is a drive straight from the manufactures or taken from an old RQDX1/2 system then you want to answer Y to this question. If this is a reformat of a drive that was already formatted on a RQDX3 system before then a N maybe answered to this question.

If a N is answered no UIT questions will appear. How ever if a Y is answered then a UIT table must be choosen according to the drive name or a UIT table built by answering about 20 questions on the drive's parameters. This questions will described in the next section.

Continue if Bad Block Information is inaccessible <N>? Y

The default to this question is N. I always answer Y. If the bad block information can not be found you still want to format your drive. For this reason I always pick Y. In most cases the manufacturing tables should be there unless you have a Proto-type drive.

Enter serial number <9 digits> ? 123456789

This question has no default. A serial number should be picked for the drive that is different then another drive on the system. This number

should be non-zero.

2.3.3 UIT TABLES

The UIT tables are stored in this program. There are 7 large data tables formed in this diagnostic that contain the drive parameters for certain DEC drives. There are only 4 RQDX3 Winchester drive manufactures. So only 4 of the tables contain any information. The others are there for future drives. If Yes is answered to the Down Line Load question then a table will be DMAed to the disk controller. The User will pick from a table the UIT number that represents the disk drive to be formatted as shown below. If the user chooses "other" then the user will have to answer all the parameter questions on his own.

Unit Information Tables listed:

Enter UIT:

UIT	Drive Name
0:	RD51
1:	RD52 part # 30 21721-02 (1 light on front panel)
2:	RD52 part # 30-23227-02 (2 lights on front panel)
3:	RD53
4:	
5:	
6:	
7:	
10:	other

Enter Unit Identifier Table (UIT) (0) ?

If you know the name of the drive then just enter the number representing the drive name. If you have a proto type drive then enter "10" representing OTHER.

Unit Information/parameter questions, used to build a UIT:

If you entered 10 in the above question you will have to answer the following questions. Note all the values entered are in octal.

XBN size (lo wrd) XBN size = 3*(1+sectors_per track) (0) value ?
XBN size (hi wrd) (0) value ?
DBN size (lo wrd) (0) value ?
DBN size (hi wrd) (0) value ?
LBN size (lo wrd) (0) value ?
LBN size (hi wrd) (0) value ?
RBN size (lo wrd) (0) value ?
RBN size (hi wrd) (0) value ?
Sectors per track (0) value ?
Surfaces per unit (0) value ?
Cylinders per unit (0) value ?
Write precomp cylinder (0) value ?
Reduce write current cylinder (0) value ?
Seek Rate (0) value ?
Use CRC or ECC (0) value ?
Number of RCT copies (0) value ?

```
289                             Media (lo wrd) (0) value ?
290                             Media (hi wrd) (0) value ?
291                             Sector Interleave (n-to-1) (0) value ?
 92                             Surface to Surface Skew (0) value ?
293                             Cylinder to Cylinder Skew (0) value ?
294                             Gap size 0 (0) value ?
295                             Gap size 1 (0) value ?
296                             Gap size 2 (0) value ?
297                             Gap size 3 (0) value ?
298                             Sync size (0) value ?
299                             MSCP cylinders per Unit (0) value ?
300                             MSCP Groups per Cylinder (0) value ?
301                             MSCP Tracks per Group (0) value ?
302                             Max allowed bad spots per surface (0) value ?
303                             Bad spot tolerance (bytes) (0) value ?
304                             --------BLANK0---------------------- (0) value ?
305                             --------BLANK1---------------------- (0) value ?
306                             --------BLANK2---------------------- (0) value ?
307                             --------BLANK3---------------------- (0) value ?
308                             --------BLANK4---------------------- (0) value ?
309                             --------BLANK5---------------------- (0) value ?
310                             --------BLANK6---------------------- (0) value ?
311                             --------BLANK7---------------------- (0) value ?
312
313
314                                     There are many questions to build a UIT table. These questions
315                                     were added mainly to help the engineers use new drives and come
316                                     up with proper parameters that would optimize the drive to the
317                                     controller. I would not suggest using this option unless you know
318                                     MSCP and disk geometry very well. It is possible to patch in the
319                                     default parameters into the table. The tables address is a UITDF:
320                                     Once the defaults are patched in, parameters can be changed very easily.
321
322
323                                      2.4      PROGRAM MESSAGES AND FORMAT COMPLETION
324
325
326
327                                     When the format finally starts a "Format Begun" message will
328                                     appear and in the end a "Format Complete" message will appear.
329                                     There may be 30+ minutes between the messages. If the extented
330                                     messages are allowed 3 "Verification Pass XXXXXX Begun" messages
331                                     will appear. Theses messages tell when the controller checks the
332                                     blocks for bad spots in the disk surface. These passes take several
333                                     minutes each and touch all the cylinders on the drive. At the end of
334                                     the format if extended messages are on a table will be printed out
335                                     reporting the results of the format. Ussually there are several bad
336                                     spots on a disk. This is very common and is NOT a mistake. These bad
337                                     blocks are revectored to new areas on the disk.
338
339                             Completion Report:
340
341                             xxx     Revectored LBNs
342                             xxx     Primary revectored LBNs
343                             xxx     Secondary/tertiary revectored LBNs
344                             xxx     Bad Blocks in the RCT area due to data errors
345                             xxx     Bad Blocks in the DBN area due to data errors
```

```
346                             xxx     Bad Blocks in the XBN area due to data errors
347                             xxx     Blocks retired on check pass
348                             ECT was not used
349                             TEST UNIT xxxx was dropped
350                             pass aborted for this unit
351                             ZRQC EQP 1
352                                 0 Cumulative errors
353
354                                 Note that every time the disk formats successfully the program
355                                 drops the UNIT. This is purposesly done so one doesn't reformat
356                                 it twice.
357
358
359                                  2.5      EXECUTION TIME
360                                 The execution time for this diagnostic varies greatly according
361                                 to the size of the drive being formatted. If an error in the
362                                 drive configuration or state such as a write protect switch
363                                 being on, an error will occur right after all the questions have
364                                 been answered. If there are no errors the formatter will take
365                                 between 5 minutes to 30 minutes depending on the drive being formatted.
366                                 A RD51 takes between 5 & 13 minutes to format depending on the way
367                                 questions are answered. A RD52 take between 10 & 25 minutes to format
368                                 and a RD53 a very long time to format. The program checks continuously
369                                 to make sure the controller is still working. If no progress is
370                                 indicated by the progress indicator a timeout error will occur. If
371                                 the disk controller goes off line for some unapparent reason the
372                                 formatter will know. Either way if one checks the light on the
373                                 Winchester to see if it is lite or check  the READY light of the drive
374                                 for a flickering light, this will tell the user that the formatter is
375                                 working.  When the formatter completes a "Format complete" message
376                                 will appear on the terminal.
377
378                                  3.      ERRORS
379
380                                 There are many types of errors possible while formatting a drive.
381                                 First the system has to be configured right. The drives have to be
382                                 jumpered right along with the disk controller. If you get an error
383                                 read the entire error message carefully. See if there is something
384                                 simple wrong such as loss and misconfigured drives before calling FS.
385                                 This is ussually the case very seldom do the drive or controller
386                                 break. So check the cables, check the jumpers, try several times and
387                                 if you still can't format then call Field Service.
388
389
390
391                             error #         Comment                     Problem
392                             0,SF0   ;unkown response
393                                     Not a DUP standard local program or Data Error in local program
394                                     execution.
395
396                             1,HRD0  ;Fatal DUP type returned
397                                     Error with Format program check detailed error message more then
398                                     likely this will be a drive error or drive configuration error.
399
400                             2,DF3   ;Can't do remote programs"
401                                     Wrong controller or bad microcode controller error.
402
```

```
                        3,SFT0  :"already active  will do an ABORT cmd"
                                Wrong controller or bad microcode controller error.

                        4,DF2   ;wrong step bit set after interrupt
                                Controller initialazation error. Controller is broken or at
                                wrong address and something is in its place.

                        5,DF1   ;controller timeout during hard init
                                Controller error, controller is slow or it can't interrupt the
                                Q bus. Controller is dead.

                        6,SFT1  ;wrong model #,wrong controller
                                This is not really an error. You are using the wrong formatter
                                program to for the wrong disk controller. It still might work
                                but no guarantees.

                        ,,DF4   ;NXM trap at controller IP address
                                Wrong configuration address of the controller check for
                                wrong jumper settings.

                        8,SF100 ;Unexpected interrupt
                                Something in system interrupting or late interrupt. This
                                could be the system clock or an interrupt from an IO port.
                                If the interrupt is at address 4,10 probable a software error
                                Try again.

                        9,DF12  ;Fatal SA error
                                Controller crashed  check detailed error message either dead
                                controller or configuration error.

                        10,DF11  Bad response packet
                                Inapropriate command or soft controller error check
                                detail message for more info.

                        11,DF13 ;no progress shown after cmd timeout
                                Format program crashed check drive & controller. The
                                error is probable in the drive. Check drive, drive cables and
                                connections. try again. If st ll a time out controller could
                                have corrupt RAM or ROM memory.

                        12,DF14 ;no iterrupt after get dust status command controller dead
                                Same as above. The error happened in the DUP monitor of the
                                controller. It was unable to respond to a command. This could
                                be a controller error


                                4.      PROGRAM DESIGN AND FLOW

                                        The program is kind of simple. There is only 1 command ring and
                                1 response ring. For every command send there is expected 1 response.
                                If the command sent times out a "Get DUST Status" command is sent to
                                check on        the controllers progress. This usually happens when the
                                actual format is being done. The rest of the commands pass information
                                back and forth from the user to the controller and back with out ever
                                t'm'ng out. This program is written according to UQSSP and DUP specs.
                                This specs can be acquired from NEWTON::ARCH$FILES:. At the start of the
```

program the INIT sequence brings the controller into the higher protocol state of running DUP commands. Once initialized the controller executed a GET DUST STATUS command to make sure the controller is in an Idle state. If idle which it should be the program asks for a program name to run. The EXECUTE LOCAL PROGRAM command is executed which should start the program into the DUP dialog loop. This dialog is described in the DUP spec. Here several SEND DATA and RECEIVE DATA commands are executed to ask questions and supply information on the success and completion of the local FORMAT program running in the RQDX3.

A pass will occur when the formatter has completed formatting all the logical units. If an error arrises the program loops until either the formatter works successfully or a the disk controller is considered broken.

5. MODIFICATION OF UIT FOR ADDITIONAL DRIVES

If the user is interested in using there own drives they may patch this diagnostic by filling some of the spare tables with there own drive parameters for there own winchester drive.,This is not suggested and DEC is not responsible for data lost on the drives because of incorrect parameters being submitted to the disk controller. The DRVTXT: location contains the ASCII data asking for the UIT number and the UIT0: thru UIT7: contain the actual UIT tables.

6.0 GLOSSARY

ZRQC follows the module name format described in the XXDP Programmer's Guide.

RQ-- Identifies the hardware and thus the module.

--C Distiguishes between two or more different modules for the same generic device. The sequence A, B, C, ETC. must be used for each additional example.

---A Specifies the module revision.

7.0 BIBLIOGRAPHY

UQSSP (NEWTON::ARCH$FILES:)
MSCP (NEWTON::ARCH$FILES:)
DUP (NEWTON::ARCH$FILES:)
DRS programmers manual (JON::disk$user1:[diaglib.drs])
XXDP programmer guide (JON::disk$user1:[diaglib.xxdp])

8.0 REVISION HISTORY

Revision B is planned to contain an autosizing routine which will size the drive instead of having the user pick the drive table.

Table of contents


```
517
518                                    )■
```

```
                                .MCALL SVC
000000                          SVC
000000                          .ENABLE ABS,AMA
                002000          .=2000
002000                          BGNMOD MOD1
002000                          POINTER BGNDU,BGNCLN,BGNPROT
002000                          HEADER ZRQC,A,0,600,0
002122                          DISPATCH 1
002126                          DESCRIPT <RQDX3 D sk Format Utility>
002160                          DEVTYPE <RD51,RD52,RD53,or Proto-type Winchester drive>

```

```
533 002236                              BGNHW DFPTBL
534 002240  172150                              .WORD   172150          ;IP address
535 002242  000154                              .WORD   154             ;Vector address
536 002244  100002                              .WORD   100002          ;Unit indentifier number rd51=1 rd52=2 rd53=3
537                                                                     ;bit 15 says it is from the Unit Identifier table
538 002246                              ENDHW
539
```

```
541 002246                              EQUALS
                                        ;
                                        ; BIT DIFINITIONS
                                        ;
100000                                  BIT15== 100000
040000                                  BIT14== 40000
020000                                  BIT13== 20000
010000                                  BIT12== 10000
004000                                  BIT11== 4000
002000                                  BIT10== 2000
001000                                  BIT09== 1000
000400                                  BIT08== 400
000200                                  BIT07== 200
000100                                  BIT06== 100
000040                                  BIT05== 40
000020                                  BIT04== 20
000010                                  BIT03== 10
000004                                  BIT02== 4
000002                                  BIT01== 2
000001                                  BIT00== 1
                                        ;
001000                                  BIT9==  BIT09
000400                                  BIT8==  BIT08
000200                                  BIT7==  BIT07
000100                                  BIT6==  BIT06
000040                                  BIT5==  BIT05
000020                                  BIT4==  BIT04
000010                                  BIT3==  BIT03
000004                                  BIT2==  BIT02
000002                                  BIT1==  BIT01
000001                                  BIT0==  BIT00
                                        ;
                                        ; EVENT FLAG DEFINITIONS
                                        ;   EF32:EF17 RESERVED FOR SUPERVISOR TO PROGRAM COMMUNICATION
                                        ;
                                        ;                                ; BIT POSITION IN SECOND STATUS WORD
000040                                  EF.START==      32.              ; (100000) START COMMAND WAS ISSUED
000037                                  EF.RESTART==    31.              ; (040000) RESTART COMMAND WAS ISSUED
000036                                  EF.CONTINUE==   30.              ; (020000) CONTINUE COMMAND WAS ISSUED
000035                                  EF.NEW==        29.              ; (010000) A NEW PASS HAS BEEN STARTED
000034                                  EF.PWR==        28.              ; (004000) A POWER-FAIL/POWER-UP OCCURRED
                                        ;
                                        ;
                                        ; PRIORITY LEVEL DEFINITIONS
                                        ;
000340                                  PRI07== 340
000300                                  PRI06== 300
000240                                  PRI05== 240
000200                                  PRI04== 200
000140                                  PRI03== 140
000100                                  PRI02== 100
000040                                  PRI01== 40
000000                                  PRI00== 0
                                        ;
                                        ;OPERATOR FLAG BITS
                                        ;
000004                                  EVL==       4
```

```
                000010          LOT==       10
                000020          ADR==       20
                000040          IDU==       40
                000100          ISR==      100
                000200          UAM==      200
                000400          BOE==      400
                001000          PNT==     1000
                002000          PRI==     2000
                004000          IXE==     4000
                010000          IBE==    10000
                020000          IER==    20000
                040000          LOE==    40000
                100000          HOE==   100000
    542                         .sbttl Literals
    543
    544                         ;+
    545                         ; Mask values to mask out specified flags
    546                         ;-
    547         000010                  UITothr = 10            ;UIT other
    548                                                         ;if UIT doesn't exist
    549
    550         000144                  UITsiz  = 144           ;UIT size
    551                         ;+
    552                         ; Misc.
    553                         ;-
    554         000004                  MaxDrv  = 4             ;Maximum Number of drives
    555         000002                  DUP.id  = bit1          ;DUP connection ID
    556         000007                  Mrqdx1  = 7.            ;model number for RQDX1
    557         000023                  Mrqdx3  = 19.           ;model number for RQDX3
    558         000001                  stdaln  = bit0
    559                         ;+
    560                         ; Opcodes for DUP commands
    561                         ;-
    562         000001                  op.gds  = 1
    563         000006                  op.abrt = 6
    564         000004                  op.gen  = 4
    565         000005                  op.rec  = 5
    566         000003                  op.elp  = 3
    567         000002                  op.esp  = 2
    568         000200                  op.end  = 200
    569                         ;+
    570                         ; Message type masks
    571                         ;-
    572         000001                  Question = 1
    573         000002                  DefQuest = 2
    574         000003                  inform   = 3
    575         000004                  terminat = 4
    576         000005                  ftlerr   = 5
    577         000006                  specl    = 6
    578
    579         177760                  type    = 177760
    580         170000                  msgnbr  = 170000
```

Macro

```
                        .sbttl  Macro
                        ;+
                        ;       Execute a GET DUST STATUS command and the check the response.
                        ;-
000000                  A=0
000001                  B=1
                        .MACRO  GETDUST         ;Execute a GET DUST STATUS command
                        B=B+1                                   ;increment the CRN number
                        gdstmp  \B              ;call variable B as if it where a number (\)
                        .ENDM

                        .MACRO  GDSTMP  B
                          list
                        GDS'B:  bit     #bit15,cmdlen+2         ;test ownership of ring make sure we own it
                                bne     GDS'B                   ;if we don't own it wait until we do
                                mov     #14.,cmdlen             ;load lenght of packet to be send
                                movb    #0,cmdlen+2             ;load msg type and credit
                                movb    #dup.id,cmdlen+3        ;load DUP connection ID
                                inc     cmdpak                  ;load new CRN
                                clr     cmdpak+2
                                clr     cmdpak+4
                                clr     cmdpak+6
                                mov     #op.gds,cmdpak+10       ;load up opcode
                                clr     cmdpak+12               ;no modifiers

                                mov     #RFD'B,@vector          ;NEW VECTOR PLACE
                                mov     #rsppak,rsprng          ;load response packet area into ring
                                mov     #cmdpak,cmdrng          ;load command packet area into ring
                                mov     #140000,RSPRNG+2        ;PORT OWNERSHIP BIT.
                                mov     #bit15,CMDRNG+2
                                jsr     pc,POLLWT               ;GO TO POLL AND WAIT ROUTINE.
                        ;*******************************************************************
                            RFD'B:                              ;INTR TO HERE.
                                mov     #intsrv,@vector         ;CHANGE VECTOR
                                jsr     pc,RSPCHK
                                                                ;GO TO ROUTINE THAT WILL CHECK ON
                                                                ;THE RESPONSE RECVD FROM THE MUT.
                                                                ;IT WILL CHECK THE CMD REF
                                                                ;NUM, THE ENDCODE AND STATUS.
                                .nlist
                        .ENDM

                        ;+
                        ;       Execute an ABORT command and then checks the response.
                        ;-
                        .MACRO  ABRT            ;Execute an ABORT command
                        B=B+1                                   ;increment the CRN number
                        abrttmp \B              ;call variable B as if it where a number (\)
                        .ENDM

                        .MACRO  ABRTTMP B
                                .list
                        ABRT'B: bit     #bit15,cmdlen+2         ;test ownership of ring make sure we own it
                                bne     ABRT'B                  ;if we don't own it wait until we do
                                mov     #14.,cmdlen             ;load lenght of packet to be send
                                movb    #0,cmdlen+2             ;load msg type and credit
                                movb    #dup.id,cmdlen+3        ;load DUP connection ID
```

```
Macro

    639                             inc     cmdpak                  ;load new CRN
    640                             clr     cmdpak+2
    641                             clr     cmdpak+4
    642                             clr     cmdpak+6
    643                             mov     #op abrt,cmdpak+10      ;load up opcode
    644                             clr     cmdpak+12               ;no modifiers
    645
    646                             mov     #RFD'B,@vector          ;NEW VECTOR PLACE
    647                             mov     #rsppak,rsprng          ;load response packet area into ring
    648                             mov     #cmdpak,cmdrng          ;load command packet area into ring
    649                             mov     #140000,RSPRNG+2        ;PORT OWNERSHIP BIT.
    650                             mov     #bit15,CMDRNG+2
    651                             jsr     pc,POLLWT               ;GO TO POLL AND WAIT ROUTINE.
    652                         ;*****************************************************************
    653                             RFD'B:                          ;INTR TO HERE.
    654                             mov     #intsrv,@vector         ;CHANGE VECTOR
    655                             jsr     pc,RSPCHK
    656                                                             ;GO TO ROUTINE THAT WILL CHECK ON
    657                                                             ;THE RESPONSE RECVD FROM THE MUT.
    658                                                             ;IT WILL CHECK THE CMD REF
    659                                                             ;NUM, THE ENDCODE AND STATUS.
    660                             .nlist
    661                         .ENDM
    662
    663
    664                         ;+
    665                         ;       Execute a Send data cmd in dup and then check the response for the proper info
    666                         ;-
    667
    668                         .MACRO  SENDDAT SPLACE,SBYTCN   ;Execute a Send Data command
    669                         B=B+1                           ;increment the CRN number
    670                         sendtmp \B,SPlace,Sbytcn        ;call variable A,B as if it where a number (\)
    671                         .ENDM
    672
    673                         .MACRO  SENDTMP B,Splace,Sbytcnt
    674                                 .list
    675                         SDT'B:  bit     #bit15,cmdlen+2         ;test ownership of ring make sure we own it
    676                                 bne     SDT'B                   ;if we don't own it wait until we do
    677                                 mov     #34,cmdlen              ;load lenght of packet to be send
    678                                 movb    #0,cmdlen+2             ;load msg type and credit
    679                                 movb    #dup.id,cmdlen+3        ;load DUP connection ID
    680                                 inc     cmdpak                  ;load new CRN
    681                                 clr     cmdpak+2
    682                                 clr     cmdpak+4
    683                                 clr     cmdpak+6
    684                                 mov     #op.sen,cmdpak+10       ;load up opcode
    685                                 clr     cmdpak+12               ;no modifiers
    686                                 mov     Sbytcnt,cmdpak+14
    687                                 clr     cmdpak+16
    688                                 mov     Splece,cmdpak+20        ;load address of buffer describtor
    689                                 clr     cmdpak+22
    690                                 clr     cmdpak+24
    691                                 clr     cmdpak+26
    692                                 clr     cmdpak+30
    693                                 clr     cmdpak+32
    694
    695                                 mov     #RFD'B,@vector          ;NEW VECTOR PLACE
```

```
Macro

    696                                     mov     #rsppak,rsprng          ;load response packet area into ring
    697                                     mov     #cmdpak,cmdrng          ;load command packet area into ring
    698                                     mov     #140000,RSPRNG+2        ;PORT OWNERSHIP BIT.
    699                                     mov     #bit15,CMDRNG+2
    700                                     jsr     pc,POLLWT               ;GO TO POLL AND WAIT ROUTINE.
    701                                 ;***************************************************************
    702                                     RFD'B:                          ;INTR TO HERE.
    703                                     mov     #intsrv,@vector         ;CHANGE VECTOR
    704                                     jsr     pc,RSPCHK
    705                                                                     ;GO TO ROUTINE THAT WILL CHECK ON
    706                                                                     ;THE RESPONSE RECVD FROM THE MUT.
    707                                                                     ;IT WILL CHECK THE CMD REF
    708                                                                     ;NUM, THE ENDCODE AND STATUS.
    709                                     .nlist
    710                                 .ENDM
    711
    712
    713                                 ;+
    714                                 ;       Execute a Receive Data command and the check the response.
    715                                 ;-
    716                                 .MACRO  RECVDAT Rplace,Rbytcnt  ;Execute a Send Data command
    717                                 B=B+1                           ;increment the CRN number
    718                                 recvtmp \B,Rplace,Rbytcnt       ;call variable A,B as if it where a number (\)
    719                                 .ENDM
    720
    721                                 .MACRO  RECVTMP B,RPlace,Rbytcnt
    722                                         .list
    723                                 RCD'B:  bit     #bit15,cmdlen+2         ;test ownership of ring make sure we own it
    724                                         bne     RCD'B                   ;if we don't own it wait until we do
    725                                         mov     #34,cmdlen              ;load lenght of packet to be send
    726                                         movb    #0,cmdlen+2             ;load msg type and credit
    727                                         movb    #dup.id,cmdlen+3        ;load DUP connection ID
    728                                         inc     cmdpak                  ;load new CRN
    729                                         clr     cmdpak+2
    730                                         clr     cmdpak+4
    731                                         clr     cmdpak+6
    732                                         mov     #op.rec,cmdpak+10       ;load up opcode
    733                                         clr     cmdpak+12               ;no modifiers
    734                                         mov     Rbytcnt,cmdpak+14
    735                                         clr     cmdpak+16
    736                                         mov     Rplace,cmdpak+20        ;load address of buffer describtor
    737                                         clr     cmdpak+22
    738                                         clr     cmdpak+24
    739                                         clr     cmdpak+26
    740                                         clr     cmdpak+30
    741                                         clr     cmdpak+32
    742
    743                                         mov     #RFD'B,@vector          ;NEW VECTOR PLACE
    744                                         mov     #rsppak,rsprng          ;load response packet area into ring
    745                                         mov     #cmdpak,cmdrng          ;load command packet area into ring
    746                                         mov     #140000,RSPRNG+2        ;PORT OWNERSHIP BIT.
    747                                         mov     #bit15,CMDRNG+2
    748                                         jsr     pc,POLLWT               ;GO TO POLL AND WAIT ROUTINE.
    749                                 ;***************************************************************
    750                                     RFD'B:                          ;INTR TO HERE.
    751                                         mov     #intsrv,@vector         ;CHANGE VECTOR
    752                                         jsr     pc,RSPCHK
```

Macro

```
                                                        ;GO TO ROUTINE THAT WILL CHECK ON
                                                        ;THE RESPONSE RECVD FROM THE MUT.
                                                        ;IT WILL CHECK THE CMD REF
                                                        ;NUM, THE ENDCODE AND STATUS.
                        .nlist
.ENDM


;+
;       Execute a Receive Data command and the check the response.
;-
.MACRO  EXLCPRG Enamadr         ;Execute a Send Data command
B=B+1                           ;increment the CRN number
elptmp  \B,Enamadr              ;call variable A,B as if it where a number (\)
.ENDM

.MACRO  ELPTMP  B,Enamadr
        .list
ELP'B:  bit     #bit15,cmdlen+2         ;test ownership of ring make sure we own it
        bne     ELP'B                   ;if we don't own it wait until we do
        mov     #22,cmdlen              ;load lenght of packet to be send
        movb    #0,cmdlen+2             ;load msg type and credit
        movb    #dup.id,cmdlen+3        ;load DUP connection ID
        inc     cmdpak                  ;load new CRN
        clr     cmdpak+2
        clr     cmdpak+4
        clr     cmdpak+6
        mov     #op.elp cmdpak+10       ;load up opcode
        mov     #stdaln,cmdpak+12       ;stand alone modifier
        mov     #6,r0                   ;6 letters transfer
        mov     #cmdpak+14,r1           ;starting address to place program name
        mov     #Enamadr,r2             ;start of Program Name
rfdj'B: movb    (r2)+,(r1)+             ;add 2 to bycnt then store
        sob     r0,rfdj'B

        mov     #RFD'B,@vector          ;NEW VECTOR PLACE
        mov     #rsppak,rsprng          ;load response packet area into ring
        mov     #cmdpak,cmdrng          ;load command packet area into ring
        mov     #140000,RSPRNG+2        ;PORT OWNERSHIP BIT.
        mov     #bit15,CMDRNG+2
        jsr     pc,POLLWT               ;GO TO POLL AND WAIT ROUTINE.
;*************************************************************************
    RFD'B:                              ;INTR TO HERE.
        mov     #intsrv,@vector         ;CHANGE VECTOR
        jsr     pc,RSPCHK
                                        ;GO TO ROUTINE THAT WILL CHECK ON
                                        ;THE RESPONSE RECVD FROM THE MUT
                                        ;IT WILL CHECK THE CMD REF
                                        ;NUM, THE ENDCODE AND STATUS.
        .nlist
.ENDM
```

Word & Buffer defintions

```
    807                                 .sbttl  Word & Buffer defintions
    808
    809 002246  000000                  LOGUNIT: .WORD                  ;logunit number
    810 002250  000000                  LOCAL:   .WORD                  ;
    811 002252  000000                  PLOC:    .WORD                  ;p table address
    812 002254  000000                  ptbl:    .WORD                  ;p table address
    813 002256  000000                  UITadr:  .word
    814
    815                                 ;+
    816                                 ; These next locations may be altered to supply the correct IP & SA address
    817                                 ; If only 1 jumper is to be placed on the MUT the locations should be filled
    818                                 ;  with addresses 177770 and 177772 respectively.
    819                                 ;-
    820 002260  000000                  IPreg:   .WORD   0              ;ADDRESS OF THE SA AND IP
    821 002262  000000                  Vector:  .word   0
    822 002264  000000                  UIN:     .word   0
    823
    824 002266                          RSP1:    .BLKW   2              ;RESPONSE PACKET LENGTH
    825 002272                          RSPPAK:  .BLKW   30.            ;RESPONSE PACKET
    826 002366                          CMDLEN:  .BLKW   2              ;COMMAND PACKET LENGTH
    827 002372                          CMDPAK:  .BLKW   20.            ;COMMAND PACKET
    828
    829 002442  000000                  CINTR:   .WORD   0              ;COMMAND INTERRUPT INDICATOR
    830 002444  000000                  RINTR:   .WORD   0              ;RESPONSE INTERRUPT INDICATOR
    831 002446  002272                  RSPRNG:  .word   rsppak         ;MESSAGE RING
    832 002450  140000                           .word   140000
    833 002452  002372                  CMDRNG:  .word   cmdpak         ;COMMAND RING
    834 002454  100000                           .word   100000
    835 002456  177777                           .WORD   -1
    836
    837 002460  000000                  LSTCRN:  .word   0              ;storage for unreturned command CRN
    838 002462  000000                  LSTCMD:  .word   0              ;storage for unreturned command opcode
    839 002464  000000                  LSTVCT:  .word   0              ;storage for unreturned command intterupt vector address
    840 002466  000000                  LOPRGI:  .word   0              ;Low word of the progress indicator
    841 002470  000000                  HIPRGI:  .word   0              ;High word of progress indicator
    842                                                                 ;data area
    843 002472      045     101     061 DATARE:  .asciz  /%A1234567890123456789012345678901234567890123456789012345678901234567890123
45
        002475      062     063     064
        002500      065     066     067
        002503      070     071     060
        002506      061     062     063
        002511      064     065     066
        002514      067     070     071
        002517      060     061     062
        002522      063     064     065
        002525      066     067     070
        002530      071     060     061
        002533      062     063     064
        002536      065     066     067
        002541      070     071     060
        002544      061     062     063
        002547      064     065     066
        002552      067     070     071
        002555      060     061     062
        002560      063     064     065
        002563      066     067     070
        002566      071     060     061
```

Word & Buffer defintions

```
        002571      062     063     064
        002574      065     066     067
        002577      070     071     060
        002602      061     062     063
        002605      064     065     066
        002610      067     070     071
        002613      060     000
    844                                 .even
    845 002616      106     117     122 PRGnam: .ascii  /FORMAT/        ;address of local format program name
        002621      115     101     124
    846 002624      000                         .byte   0               ;null for asciz
    847                                 .even
    848
```

```
                                .sbttl  DISK UNIT INFORMATION TABLE
                                ;+
                                ; The following tables are made up of disk drive parameters which will be
                                ; feed to the FORMAT controller local program which will then use the
                                ; information to format tne drives.
                                ;-
         003000                 .=3000
003000                          UIT0:
                                ;+
                                ;       Unit Information table RD51 Seagate
                                ;-
                                                        ;/*Top of Unit Information table (UIT)
003000  000071                          .word   57.     ;/XBN size (lo wrd) XBN size = 3*(1+sectors_per_track)/
003002  000000                          .word   0       ;/XBN size (hi wrd)/
003004  000127                          .word   87.     ;/DBN size (lo wrd)/
003006  000000                          .word   0       ;/DBN size (hi wrd)/
003010  052360                          .word   21744.  ;/LBN size (lo wrd)/
003012  000000                          .word   0       ;/LBN size (hi wrd)/
003014  000220                          .word   144.    ;/RBN size (lo wrd)/
003016  000000                          .word   0       ;/RBN size (hi wrd)/
003020  000022                          .word   18.     ;/Sectors per track/
003022  000004                          .word   4.      ;/Surfaces per unit/
003024  000462                          .word   306.    ;/Cylinders per unit/
003026  000156                           word   110.    ;/Write precomp cylinder/
003030  000462                          .word   306.    ;/Reduce write current cylinder /
003032  000000                          .word   0       ;/Seek Rate/
003034  000001                          .word   1       ;/Use CRC or ECC/
003036  000044                          .word   36.     ;/RCT Size/
003040  000004                          .word   4.      ;/Number of RCT copies/
003042  040063                          .word   ^B0100000000110011 ;^H4033;/Media (lo wrd)/
003044  022544                          .word   ^B0010010101100100 ;^H2564;/Media (hi wrd)/
003046  000001                          .word   1       ;/Sector Interleave (n to-1)/
003050  000002                          .word   2       ;/Surface to Surface Skew/
003052  000000                          .word   0       ;/Cylinder to Cylinder Skew/
003054  000020                          .word   16.     ;/Gap size 0/
003056  000020                          .word   16.     ;/Gap size 1/
003060  000010                          .word   8.      ;/Gap size 2/
003062  000043                          .word   35.     ;/Gap size 3/
003064  000015                          .word   13.     ;/Sync size/
003066  000001                          .word   1       ;/MSCP cylinders per Unit/
003070  000001                          .word   1       ;/MSCP Groups per Cylinder/
003072  000001                          .word   1       ;/MSCP Tracks per Group/
003074  000020                          .word   20      ;/Max allowed bad spots per surface/
003076  000120                          .word   120     ;/Bad spot tolerance (bytes)/
003100  000000                          .word   0       ;/_______filler0------ --- ----------/
003102  000000                          .word   0       ;/_______filler1---------------------/
003104  000000                          .word   0       ;/_______filler2---------------------/
003106  000000                          .word   0       ;/_______filler3---------------------/
003110  000000                          .word   0       ;/_______filler4---------- ----------/
003112  000000                          .word   0       ;/_______filler5------------ --------/
003114  000000                          .word   0       ;/_______filler6---------------------/
003116  000000                          .word   0       ;/_______filler7---------------------/

         003144                 .=3000+ UITsiz
003144                          UIT1:
                                ;+
                                ;       Unit Information table    RD52 Quantum drive
```

DISK UNIT INFORMATION TABLE

```
                                        ;-
                                                        ;/*Top of Unit Information table (UIT)
003144  000066                  .word   54.            ;/XBN size (lo wrd) XBN size = 3*(1+sectors_per_track)/
003146  000000                  .word   0              ;/XBN size (hi wrd)/
003150  000100                  .word   64.            ;/DBN size (lo wrd)/
003152  000000                  .word   0              ;/DBN size (hi wrd)/
003154  166114                  .word   60492.         ;/LBN size (lo wrd)/
003156  000000                  .word   0              ;/LBN size (hi wrd)/
003160  000250                  .word   168.           ;/RBN size (lo wrd)/
003162  000000                  .word   0              ;/RBN size (hi wrd)/
003164  000021                  .word   17.            ;/Sectors per track/
003166  000010                  .word   8.             ;/Surfaces per unit/
003170  001000                  .word   512.           ;/Cylinders per unit/
003172  000400                  .word   256.           ;/Write precomp cylinder/
003174  001000                  .word   512.           ;/Reduce write current cylinder /
003176  000000                  .word   0              ;/Seek Rate/
003200  000001                  .word   1              ;/Use CRC or ECC/
003202  000004                  .word   4              ;/RCT Size/
003204  000003                  .word   3              ;/Number of RCT copies/
003206  040064                  .word   ^B0100000000110100 ;^H4034;/Media (lo wrd)/
003210  022544                  .word   ^B0010010101100100 ;^H2564./Media (hi wrd)/
003212  000001                  .word   1              ;/Sector Interleave (n-to-1)/
003214  000002                  .word   2              ;/Surface to Surface Skew/
003216  000000                  .word   0              ;/Cylinder to Cylinder Skew/
003220  000020                  .word   16.            ;/Gap size 0/
003222  000020                  .word   16.            ;/Gap size 1/
003224  000010                  .word   8.             ;/Gap size 2/
003226  000043                  .word   35.            ;/Gap size 3/
003230  000015                  .word   13.            ;/Sync size/
003232  000001                  .word   1              ;/MSCP cylinders per Unit/
003234  000001                  .word   1              ;/MSCP Groups per Cylinder/
003236  000001                  .word   1              ;/MSCP Tracks per Group/
003240  000020                  .word   20             ;/Max allowed bad spots per surface/
003242  000120                  .word   120            ;/Bad spot tolerance (bytes)/
003244  000000                  .word   0              ;/_______filler0--------------------/
003246  000000                  .word   0              ;/_______filler1--------------------/
003250  000000                  .word   0              ;/_______filler2--------------------/
003252  000000                  .word   0              ;/_______filler3---- ---------------/
003254  000000                  .word   0              ;/_______filler4--------------------/
003256  000000                  .word   0              ;/_______filler5---------- ---------/
003260  000000                  .word   0              ;/_______filler6---------- ---------/
003262  000000                  .word   0              ;/_______filler7-------- ------------/

        003310          .=3000+UITsiz+UITsiz
003310                  UIT2:
                        ;+
                        ;       Unit Information table  RD52 Atasi
                        ;-
                                                        ;/*Top of Unit Information table (UIT)
003310  000066                  .word   54.            ;/XBN size (lo wrd) XBN size = 3*(1+sectors_per_track)/
003312  000000                  .word   0              ;/XBN size (hi wrd)/
003314  000100                  .word   64.            ;/DBN size (lo wrd)/
003316  000000                  .word   0              ;/DBN size (hi wrd)/
003320  166114                  .word   60492.         ;/LBN size (lo wrd)/
003322  000000                  .word   0              ;/LBN size (hi wrd)/
003324  000250                  .word   168.           ;/RBN size (lo wrd)/
003326  000000                  .word   0              ;/RBN size (hi wrd)/
```

DISK UNIT INFORMATION TABLE

```
 964 003330  000021                .word   17.             ;/Sectors per track/
 965 003332  000007                .word   7.              ;/Surfaces per unit/
 966 003334  001205                .word   645.            ;/Cylinders per unit/
 967 003336  000500                .word   320.            ;/Write precomp cylinder/
 968 003340  001205                .word   645.            ;/Reduce write current cylinder /
 969 003342  000000                .word   0               ;/Seek Rate/
 970 003344  000001                .word   1               ;/Use CRC or ECC/
 971 003346  000004                .word   4               ;/RCT Size/
 972 003350  000003                .word   3               ;/Number of RCT copies/
 973 003352  040064                .word   ↑B0100000000110100 ;↑H4034;/Media (lo wrd)/
 974 003354  022544                .word   ↑B0010010101100100 ;↑H2564;/Media (hi wrd)/
 975 003356  000001                .word   1               ;/Sector Interleave (n-to-1)/
 976 003360  000002                .word   2               ;/Surface to Surface Skew/
 977 003362  000000                .word   0               ;/Cylinder to Cylinder Skew/
 978 003364  000020                .word   16.             ;/Gap size 0/
 979 003366  000020                .word   16.             ;/Gap size 1/
 980 003370  000010                .word   8.              ;/Gap size 2/
 981 003372  000043                .word   35.             ;/Gap size 3/
 982 003374  000015                .word   13.             ;/Sync size/
 983 003376  000001                .word   1               ;/MSCP cylinders per Unit/
 984 003400  000001                .word   1               ;/MSCP Groups per Cylinder/
 985 003402  000001                .word   1               ;/MSCP Tracks per Group/
 986 003404  000020                .word   20              ;/Max allowed bad spots per surface/
 987 003406  000120                .word   120             ;/Bad spot tolerance (bytes)/
 988 003410  000000                .word   0               ;/_______filler0--------------------/
 989 003412  000000                .word   0               ;/_______filler1------- ----- -------/
 990 003414  000000                .word   0               ;/_______filler2--------------------/
 991 003416  000000                .word   0               ;/_______filler3-------------- -----/
 992 003420  000000                .word   0               ;/_______filler4--------------------/
 993 003422  000000                .word   0               ;/_______filler5--------------------/
 994 003424  000000                .word   0               ;/_ _____filler6------ ---------- --/
 995 003426  000000                .word   0               ;/_______filler7---------------- ----/
 996
 997         003454        .=3000+UITsiz+UITsiz+UITsiz
 998 003454                UIT3:
 999                       ;+
1000                       ;       Unit Information table  RD53 Micropolis
1001                       ;-
1002                                                       ;/*Top of Unit Information table (UIT)
1003 003454  000066                .word   54.             ;/XBN size (lo wrd) XBN size = 3*(1+sectors_per_track)/
1004 003456  000000                .word   0               ;/XBN size (hi wrd)/
1005 003460  000100                .word   64.             ;/DBN size (lo wrd)/
1006 003462  000000                .word   0               ;/DBN size (hi wrd)/
1007 003464  016677                .word   016677          ;/LBN size (lo wrd)/
1008 003466  000002                .word   2               ;/LBN size (hi wrd)/
1009 003470  000524                .word   340.            ;/RBN size (lo wrd)/
1010 003472  000000                .word   0               ;/RBN size (hi wrd)/
1011 003474  000021                 word   17.             ;/Sectors per track/
1012 003476  000010                .word   8.              ;/Surfaces per unit/
1013 003500  002000                .word   1024.           ;/Cylinders per unit/
1014 003502  002000                .word   1024.           ;/Write precomp cylinder/
1015 003504  002000                .word   1024.           ;/Reduce write current cylinder /
1016 003506  000000                .word   0               ;/Seek Rate/
1017 003510  000001                .word   1               ;/Use CRC or ECC/
1018 003512  000005                .word   5               ;/RCT Size/
1019 003514  000003                .word   3               ;/Number of RCT copies/
1020 003516  040065                .word   ↑B0100000000110101 ;↑H4035;/Media (lo wrd)
```

DISK UNIT INFORMATION TABLE

```
1021 003520  022544                    .word   ↑B0010010101100100 ;↑H2564;/Media (hi wrd)/
1022 003522  000001                    .word   1               ;/Sector Interleave (n-to-1)/
1023 003524  000002                    .word   2               ;/Surface to Surface Skew/
1024 003526  000000                    .word   0               ;/Cylinder to Cylinder Skew/
1025 003530  000020                    .word   16.             ;/Gap size 0/
1026 003532  000020                    .word   16.             ;/Gap size 1/
1027 003534  000010                    .word   8.              ;/Gap size 2/
1028 003536  000043                    .word   35.             ;/Gap size 3/
1029 003540  000015                    .word   13.             ;/Sync size/
1030 003542  000001                    .word   1               ;/MSCP cylinders per Unit/
1031 003544  000001                    .word   1               ;/MSCP Groups per Cylinder/
1032 003546  000001                    .word   1               ;/MSCP Tracks per Group/
1033 003550  000020                    .word   20              ;/Max allowed bad spots per surface/
1034 003552  000120                    .word   120             ;/Bad spot tolerance (bytes)/
1035 003554  000000                    .word   0               ;/_______filler0-- ----------------/
1036 003556  000000                    .word   0               ;/_______filler1-------------------/
1037 003560  000000                    .word   0               ;/_______filler2--------  ------ -- /
1038 003562  000000                    .word   0               ;/_______filler3---------- ------- -/
1039 003564  000000                    .word   0               ;/_______filler4------------ -------/
1040 003566  000000                    .word   0               ;/_______filler5-------------------/
1041 003570  000000                    .word   0               ;/_______filler6----------- ------ -/
1042 003572  000000                    .word   0               ;/_______filler7 ------ ---  --------/
1043
1044         003620             .=3000+UITsiz+UITsiz+UITsiz+UITsiz
1045 003620                     UIT4:
1046                            ;+
1047                            ;       Unit Information table
1048                            ;-
1049
1050 003620  000000                                    ;/*Top of Unit Information table (UIT)
                                .word   0               ;/XBN size (lo wrd) XBN size = 3*(1+sectors_per_track)/
1051 003622  000000             .word   0               ;/XBN size (hi wrd)/
1052 003624  000000             .word   0               ;/DBN size (lo wrd)/
1053 003626  000000             .word   0               ;/DBN size (hi wrd)/
1054 003630  000000             .word   0               ;/LBN size (lo wrd)/
1055 003632  000000             .word   0               ;/LBN size (hi wrd)/
1056 003634  000000             .word   0               ;/RBN size (lo wrd)/
1057 003636  000000             .word   0               ;/RBN size (hi wrd)/
1058 003640  000000             .word   0               ;/Sectors per track/
1059 003642  000000             .word   0               ;/Surfaces per unit/
1060 003644  000000             .word   0               ;/Cylinders per unit/
1061 003646  000000             .word   0               ;/Write precomp cylinder/
1062 003650  000000             .word   0               ;/Reduce write current cylinder /
1063 003652  000000             .word   0               ;/Seek Rate/
1064 003654  000000             .word   0               ;/Use CRC or ECC/
1065 003656  000000             .word   0               ;/RCT Size/
1066 003660  000000             .word   0               ;/Number of RCT copies/
1067 003662  000000             .word   0               ;/Media (lo wrd)/
1068 003664  000000             .word   0               ;/Media (hi wrd)/
1069 003666  000000             .word   0               ;/Sector Interleave (n-to-1)/
1070 003670  000000             .word   0               ;/Surface to Surface Skew/
1071 003672  000000             .word   0               ;/Cylinder to Cylinder Skew/
1072 003674  000000             .word   0               ;/Gap size 0/
1073 003676  000000             .word   0               ;/Gap size 1/
1074 003700  000000             .word   0               ;/Gap size 2/
1075 003702  000000             .word   0               ;/Gap size 3/
1076 003704  000000             .word   0               ;/Sync size/
1077 003706  000000             .word   0               ;/MSCP cylinders per Unit/
```

DISK UNIT INFORMATION TABLE

```
1078 003710  000000                    .word   0               ;/MSCP Groups per Cylinder/
1079 003712  000000                    .word   0               ;/MSCP Tracks per Group/
1080 003714  000000                    .word   0               ;/Max allowed bad spots per surface/
1081 003716  000000                    .word   0               ;/Bad spot tolerance (bytes)/
1082 003720  000000                    .word   0               ;/_____filler0_____/
1083 003722  000000                    .word   0               ;/_____filler1_____/
1084 003724  000000                    .word   0               ;/_____filler2_____/
1085 003726  000000                    .word   0               ;/_____filler3_____/
1086 003730  000000                    .word   0               ;/_____filler4_____/
1087 003732  000000                    .word   0               ;/_____filler5_____/
1088 003734  000000                    .word   0               ;/_____filler6_____/
1089 003736  000000                    .word   0               ;/_____filler7_____/
1090
1091         003764             .=3000+UITsiz+UITsiz+UITsiz+UITsiz+UITsiz
1092 003764                     UIT5:
1093                             ;+
1094                             ;       Unit Information table
1095                             ;-
1096                                                     ;/*Top of Unit Information table (UIT)
1097 003764  000000                    .word   0               ;/XBN size (lo wrd) XBN size = 3*(1+sectors_per_track)/
1098 003766  000000                    .word   0               ;/XBN size (hi wrd)/
1099 003770  000000                    .word   0               ;/DBN size (lo wrd)/
1100 003772  000000                    .word   0               ;/DBN size (hi wrd)/
1101 003774  000000                    .word   0               ;/LBN size (lo wrd)/
1102 003776  000000                    .word   0               ;/LBN size (hi wrd)/
1103 004000  000000                    .word   0               ;/RBN size (lo wrd)/
1104 004002  000000                    .word   0               ;/RBN size (hi wrd)/
1105 004004  000000                    .word   0               ;/Sectors per track/
1106 004006  000000                    .word   0               ;/Surfaces per unit/
1107 004010  000000                    .word   0               ;/Cylinders per unit/
1108 004012  000000                    word    0               ;/Write precomp cylinder/
1109 004014  000000                    .word   0               ;/Reduce write current cylinder /
1110 004016  000000                    .word   0               ;/Seek Rate/
1111 004020  000000                    .word   0               ;/Use CRC or ECC/
1112 004022  000000                    .word   0               ;/RCT Size/
1113 004024  000000                    .word   0               ;/Number of RCT copies/
1114 004026  000000                    .word   0               ;/Media (lo wrd)/
1115 004030  000000                    .word   0               ;/Media (hi wrd)/
1116 004032  000000                    .word   0               ;/Sector Interleave (n to-1)/
1117 004034  000000                    .word   0               ;/Surface to Surface Skew/
1118 004036  000000                    .word   0               ;/Cylinder to Cylinder Skew/
1119 004040  000000                    .word   0               ;/Gap size 0/
1120 004042  000000                    .word   0               ;/Gap size 1/
1121 004044  000000                    .word   0               ;/Gap size 2/
1122 004046  000000                    .word   0               ;/Gap size 3/
1123 004050  000000                    .word   0               ;/Sync size/
1124 004052  000000                    .word   0               ;/MSCP cylinders per Unit/
1125 004054  000000                    .word   0               ;/MSCP Groups per Cylinder/
1126 004056  000000                    .word   0               ;/MSCP Tracks per Group/
1127 004060  000000                    .word   0               ;/Max allowed bad sectors per surface/
1128 004062  000000                    .word   0               ;/Bad spot tolerance (bytes)/
1129 004064  000000                    .word   0               ;/_____filler0_____/
1130 004066  000000                    .word   0               ;/_____filler1_____/
1131 004070  000000                    .word   0               ;/_____filler2_____/
1132 004072  000000                    .word   0               ;/_____filler3_____/
1133 004074  000000                    word    0               ;/_____filler4_____/
1134 004076  000000                    word    0               ;/_____filler5_____/
```

DISK UNIT INFORMATION TABLE

```
1135 004100  000000                    .word   0               ;/_______filler6--------------------/
1136 004102  000000                    .word   0               ;/_______filler7-------------------  ./
1137
1138         004130                .=3000+UITsiz+UITsiz+UITsiz+UITsiz+UITsiz+UITsiz
1139 004130                        UIT6:
1140                               ;+
1141                               ;       Unit Information table
1142                               ;-
1143                                                          ;/*Top of Unit Information table (UIT)
1144 004130  000000                    .word   0               ;/XBN size (lo wrd) XBN size = 3*(1+sectors_per_track)/
1145 004132  000000                    .word   0               ;/XBN size (hi wrd)/
1146 004134  000000                    .word   0               ;/DBN size (lo wrd)/
1147 004136  000000                    .word   0               ;/DBN size (hi wrd)/
1148 004140  000000                    .word   0               ;/LBN size (lo wrd)/
1149 004142  000000                    .word   0               ;/LBN size (hi wrd)/
1150 004144  000000                    .word   0               ;/RBN size (lo wrd)/
1151 004146  000000                    .word   0               ;/RBN size (hi wrd)/
1152 004150  000000                    .word   0               ;/Sectors per track/
1153 004152  000000                    .word   0               ;/Surfaces per unit/
1154 004154  000000                    .word   0               ;/Cylinders per unit/
1155 004156  000000                    .word   0               ;/Write precomp cylinder/
1156 004160  000000                    .word   0               ;/Reduce write current cylinder /
1157 004162  000000                    .word   0               ;/Seek Rate/
1158 004164  000000                    .word   0               ;/Use CRC or ECC/
1159 004166  000000                    .word   0               ;/RCT Size/
1160 004170  000000                    .word   0               ;/Number of RCT copies/
1161 004172  000000                    .word   0               ;/Media (lo wrd)/
1162 004174  000000                    .word   0               ;/Media (hi wrd)/
1163 004176  000000                    .word   0               ;/Sector Interleave (n-to-1)/
1164 004200  000000                    .word   0               ;/Surface to Surface Skew/
1165 004202  000000                    .word   0               ;/Cylinder to Cylinder Skew/
1166 004204  000000                    .word   0               ;/Gap size 0/
1167 004206  000000                    .word   0               ;/Gap size 1/
1168 004210  000000                    .word   0               ;/Gap size 2/
1169 004212  000000                    .word   0               ;/Gap size 3/
1170 004214  000000                    .word   0               ;/Sync size/
1171 004216  000000                    .word   0               ;/MSCP cylinders per Unit/
1172 004220  000000                    .word   0               ;/MSCP Groups per Cylinder/
1173 004222  000000                    .word   0               ;/MSCP Tracks per Group/
1174 004224  000000                    .word   0               ;/Max allowed bad spots per surface/
1175 004226  000000                    .word   0               ;/Bad spot tolerance (bytes)/
1176 004230  000000                    .word   0               ;/_______filler)--------------------/
1177 004232  000000                    .word   0               ;/_______filler1--------------------/
1178 004234  000000                    .word   0               ;/_______filler2--------------------/
1179 004236  000000                    .word   0               ;/_______filler3--------------------/
1180 004240  000000                    .word   0               ;/_______filler4--------------------/
1181 004242  000000                    .word   0               ;/_______filler5--------------------/
1182 004244  000000                    .word   0               ;/_______filler6--------------------/
1183 004246  000000                    .word   0               ;/_______filler7----------------
1184
1185
1186         004274                .=3000+UITsiz+UITsiz+UITsiz+UITsiz+UITsiz+UITsiz+UITsiz
1187 004274                        UIT7:
1188                               ;+
1189                               ;       Unit Information table
1190                               ;-
1191                                                          ;/*Top of Unit Information table (UIT)
```

DISK UNIT INFORMATION TABLE

```
1192 004274  000000                    .word   0       ;/XBN size (lo wrd) XBN size = 3*(1+sectors_per_track)/
1193 004276  000000                    .word   0       ;/XBN size (hi wrd)/
1194 004300  000000                    .word   0       ;/DBN size (lo wrd)/
1195 004302  000000                    .word   0       ;/DBN size (hi wrd)/
1196 004304  000000                    .word   0       ;/LBN size (lo wrd)/
1197 004306  000000                    .word   0       ;/LBN size (hi wrd)/
1198 004310  000000                    .word   0       ;/RBN size (lo wrd)/
1199 004312  000000                    .word   0       ;/RBN size (hi wrd)/
1200 004314  000000                    .word   0       ;/Sectors per track/
1201 004316  000000                    .word   0       ;/Surfaces per unit/
1202 004320  000000                    .word   0       ;/Cylinders per unit/
1203 004322  000000                    .word   0       ;/Write precomp cylinder/
1204 004324  000000                    .word   0       ;/Reduce write current cylinder /
1205 004326  000000                    .word   0       ;/Seek Rate/
1206 004330  000000                    .word   0       ;/Use CRC or ECC/
1207 004332  000000                    .word   0       ;/RCT Size/
1208 004334  000000                    .word   0       ;/Number of RCT copies/
1209 004336  000000                    .word   0       ;/Media (lo wrd)/
1210 004340  000000                    .word   0       ;/Media (hi wrd)/
1211 004342  000000                    .word   0       ;/Sector Interleave (n-to-1)/
1212 004344  000000                    .word   0       ;/Surface to Surface Skew/
1213 004346  000000                    .word   0       ;/Cylinder to Cylinder Skew/
1214 004350  000000                    .word   0       ;/Gap size 0/
1215 004352  000000                    .word   0       ;/Gap size 1/
1216 004354  000000                    .word   0       ;/Gap size 2/
1217 004356  000000                    .word   0       ;/Gap size 3/
1218 004360  000000                    .word   0       ;/Sync size/
1219 004362  000000                    .word   0       ;/MSCP cylinders per Unit/
1220 004364  000000                    .word   0       ;/MSCP Groups per Cylinder/
1221 004366  000000                    .word   0       ;/MSCP Tracks per Group/
1222 004370  000000                    .word   0       ;/Max allowed bad spots per surface/
1223 004372  000000                    .word   0       ;/Bad spot tolerance (bytes)/
1224 004374  000000                    .word   0       ;/_______filler0--------------------/
1225 004376  000000                    .word   0       ;/_______filler1--------------------/
1226 004400  000000                    .word   0       ;/_______filler2---------- ---------/
1227 004402  000000                    .word   0       ;/_______filler3--------------------/
1228 004404  000000                    .word   0       ;/_______filler4--------------------/
1229 004406  000000                    .word   0       ;/_______filler5------------ -------/
1230 004410  000000                    .word   0       ;/_______filler6--------------------/
1231 004412  000000                    .word   0       ;/_______filler7------------ -------/
1232
1233         004440            .=3000+UITsiz+UITsiz+UITsiz+UITsiz+UITsiz+UITsiz+UITsiz+UITsiz
1234 004440                    UITdf:
1235                           ;+
1236                           ;       DEFAULT Unit Information table
1237                           ;-
1238                                                   ;/*Top of Unit Information table (UIT)
1239 004440  000066                    .word   54.     ;/XBN size (lo wrd) XBN size = 3*(1+sectors_per_track)/
1240 004442  000000                    .word   0       ;/XBN size (hi wrd)/
1241 004444  000100                    .word   64.     ;/DBN size (lo wrd)/
1242 004446  000000                    .word   0       ;/DBN size (hi wrd)/
1243 004450  024374                    .word   10492.  ;/LBN size (lo wrd)/
1244 004452  000001                    .word   1       ;/LBN size (hi wrd)/
1245 004454  000250                    .word   168.    ;/RBN size (lo wrd)/
1246 004456  000000                    .word   0       ;/RBN size (hi wrd)/
1247 004460  000021                    .word   17.     ;/Sectors per track/
1248 004462  000010                    .word   8.      ;/Surfaces per unit/
```

DISK UNIT INFORMATION TABLE

```
   1249 004464  001000                .word   512.            ;/Cylinders per unit/
   1250 004466  000400                .word   256.            ;/Write precomp cylinder/
   1251 004470  001000                .word   512.            ;/Reduce write current cylinder /
   1252 004472  000000                .word   0               ;/Seek Rate/
   1253 004474  000001                .word   1               ;/Use CRC or ECC/
   1254 004476  000004                .word   4               ;/RCT Size/
   1255 004500  000003                .word   3               ;/Number of RCT copies/
   1256 004502  040064                .word   ^B0100000000110100 ;^H4034;/Media (lo wrd)/
   1257 004504  022544                .word   ^B0010010101100100 ;^H2564;/Media (hi wrd)/
   1258 004506  000001                .word   1               ;/Sector Interleave (n-to-1)/
   1259 004510  000002                .word   2               ;/Surface to Surface Skew/
   1260 004512  000000                .word   0               ;/Cylinder to Cylinder Skew/
   1261 004514  000020                .word   16.             ;/Gap size 0/
   1262 004516  000020                .word   16.             ;/Gap size 1/
   1263 004520  000010                .word   8.              ;/Gap size 2/
   1264 004522  000043                .word   35.             ;/Gap size 3/
   1265 004524  000015                .word   13.             ;/Sync size/
   1266 004526  000001                .word   1               ;/MSCP cylinders per Unit/
   1267 004530  000001                .word   1               ;/MSCP Groups per Cylinder/
   1268 004532  000001                .word   1               ;/MSCP Tracks per Group/
   1269 004534  000020                .word   20              ;/Max allowed bad spots per surface/
   1270 004536  000120                .word   120             ;/Bad spot tolerance (bytes)/
   1271 004540  000000                .word   0               ;/_______filler0---------------------/
   1272 004542  000000                .word   0               ;/_______filler1---------------------/
   1273 004544  000000                .word   0               ;/_______filler2---------------------/
   1274 004546  000000                .word   0               ;/_______filler3---------------------/
   1275 004550  000000                .word   0               ;/_______filler4---------------------/
   1276 004552  000000                .word   0               ;/_______filler5---------------------/
   1277 004554  000000                .word   0               ;/_______filler6---------------------/
   1278 004556  000000                .word   0               ;/_______filler7---------------------/
   1279
```

DISK UNIT INFORMATION TABLE

```
1281          .nlist  bin
1282          .sbttl DISK PARAMETER QUESTIONS
1283          ;*
1284          ; P table Questions
1285          ;-
1286 004560  IP.adr:  .ASCIZ  /IP address/
1287 004573  Vec.adr: .ASCIZ  /Vector address/
1288
1289 004612  DrvTxa: .asciz /%N%AUIT    Drive Name/
1290 004640  DrvTxb: .asciz /%N%A_____________________________________________________/
1291 004735  DrvTx0: .asciz /%N%A 0:    RD51 /
1292 004756  DrvTx1: .asciz /%N%A 1:    RD52 part # 30-21721-02 (1 light on front panel)/
1293 005052  DrvTx2: .asciz /%N%A 2:    RD52 part # 30-23227-02 (2 lights on front panel)/
1294 005147  DrvTx3: .asciz /%N%A 3:    RD53 /
1295 005170  DrvTx4: .asciz /%N%A 4:                                                  /
1296 005265  DrvTx5: .asciz /%N%A 5:                                                  /
1297 005362  DrvTx6: .asciz /%N%A 6:                                                  /
1298 005457  DrvTx7: .asciz /%N%A 7:                                                  /
1299 005554  DrvTxc: .asciz /%N%A10:    other%N%N/
1300
1301
1302 005601  Unt.nbr: .ASCIZ  /Enter Unit Identifier Table (UIT)/
1303 005643  ask.prg: .ASCIZ  /What local program do you want to run/
1304
1305
1306          ;/*Top of Unit Information table (UIT)
1307 005711  TBQ0:    .ASCIZ  /XBN size (lo wrd) XBN size = 3*(1+sectors_per_track)/
1308 005776  TBQ1:    .ASCIZ  /XBN size (hi wrd)/
1309 006020  TBQ2:    .ASCIZ  /DBN size (lo wrd)/
1310 006042  TBQ3:    .ASCIZ  /DBN size (hi wrd)/
1311 006064  TBQ4:    .ASCIZ  /LBN size (lo wrd)/
1312 006106  TBQ5:    .ASCIZ  /LBN size (hi wrd)/
1313 006130  TBQ6:    .ASCIZ  /RBN size (lo wrd)/
1314 006152  TBQ7:    .ASCIZ  /RBN size (hi wrd)/
1315 006174  TBQ8:    .ASCIZ  /Sectors per track/
1316 006216  TBQ9:    .ASCIZ  /Surfaces per unit/
1317 006240  TBQ10:   .ASCIZ  /Cylinders per unit/
1318 006263  TBQ11:   .ASCIZ  /Write precomp cylinder/
1319 006312  TBQ12:   .ASCIZ  /Reduce write current cylinder /
1320 006351  TBQ13:   .ASCIZ  /Seek Rate/
1321 006363  TBQ14:   .ASCIZ  /Use CRC or ECC/
1322 006402  TBQ15:   .ASCIZ  /RCT Size/
1323 006413  TBQ16:   .ASCIZ  /Number of RCT copies/
1324 006440  TBQ17:   .ASCIZ  /Media (lo wrd)/
1325 006457  TBQ18:   .ASCIZ  /Media (hi wrd)/
1326 006476  TBQ19:   .ASCIZ  /Sector Interleave (n-to-1)/
1327 006531  TBQ20:   .ASCIZ  /Surface to Surface Skew/
1328 006561  TBQ21:   .ASCIZ  /Cylinder to Cylinder Skew/
1329 006613  TBQ22:   .ASCIZ  /Gap size 0/
1330 006626  TBQ23:   .ASCIZ  /Gap size 1/
1331 006641  TBQ24:   .ASCIZ  /Gap size 2/
1332 006654  TBQ25:   .ASCIZ  /Gap size 3/
1333 006667  TBQ26:   .ASCIZ  /Sync size/
1334 006701  TBQ28:   .ASCIZ  /MSCP cylinders per Unit/
1335 006731  TBQ29:   .ASCIZ  /MSCP Groups per Cylinder/
1336 006762  TBQ30:   .ASCIZ  /MSCP Tracks per Group/
1337 007010  TBQ31:   .ASCIZ  /Max allowed bad spots per surface/
```

DISK PARAMETER QUESTIONS SEQ 0031

```
1338 007052  TBQ32:   .ASCIZ  /Bad spot tolerance (bytes)/
1339 007105  TBQ33:   .ASCIZ  /________BLANK0_____________________/
1340 007152  TBQ34:   .ASCIZ  /________BLANK1_____________________/
1341 007217  TBQ35:   .ASCIZ  /________BLANK2_____________________/
1342 007264  TBQ36:   .ASCIZ  /________BLANK3_____________________/
1343 007331  TBQ37:   .ASCIZ  /________BLANK4_____________________/
1344 007376  TBQ38:   .ASCIZ  /________BLANK5_____________________/
1345 007443  TBQ39:   .ASCIZ  /________BLANK6_____________________/
1346 007510  TBQ40:   .ASCIZ  /_____ __BLANK7_____________________/
1347
1348 007555  DF1:     .ASCIZ  /Controller Initialization Timeout/
1349 007617  DF2:     .ASCIZ  /Controller never advanced to next step/
1350 007666  DF3:     .ASCIZ  /Controller can not execute local programs or non STD DUP dialog program/
1351 007776  DF4:     .ASCIZ  /NXM Trap at controllers IP address/
1352         ;DF10:   .ASCIZ  /No Interrupt occurred after SA polled/
1353 010041  DF11:    .ASCIZ  /Bad Response Packet returned/
1354 010076  DF12:    .ASCIZ  /Fatal SA error ctlr offline/
1355 010132  DF13:    .ASCIZ  /No progress shown after a cmd had timed out/
1356 010206  DF14:    .ASCIZ  /GET DUST CMD time_out after another CMD time_out/
1357 010267  DF15:    .ASCIZ  /%N%AFatal error was reported when running local program/
1358 010357  DF16:    .ASCIZ  /%N%AA Special was reported when running local program don't know how to handle it/
1359 010501  SF0:     .ASCIZ  /DUP protocol Error, unexpected message/
1360 010550  SF1:     .ASCIZ  /%N%ASYSTEM is NOT in manual mode/
1361 010611  SF100:   .ASCIZ  /Unexpected or delayed Controller Interrupt/
1362 010664  HRD0:    .ASCIZ  /Fatal Format error/
1363 010707  SFT0:    .ASCIZ  /Controller in an unexpected ACTIVE state/
1364 010760  SFT1:    .ASCIZ  /Wrong Model Number on controller/
1365 011021  PB0:     .ASCIZ  /%N%AModel # listed %06/
1366 011050  PB1:     .ASCIZ  /%N%AExpected SA step bit %06%A,Received in SA %06/
1367 011132  PB3:     .ASCIZ  /%N%AAsking for Format Parameter table/
1368 011200  PB4:     .ASCIZ  /%N%AReceived valid Format Parameter table/
1369 011252  PB5:     .ASCIZ  /%N%AOn UNIT %06%A, %06 Bad Blks were found during Format/
1370 011343  PB6:     .ASCIZ  /%N%AOn UNIT %06%A, %06 Bad Blks were found during Verify pass %06/
1371 011445  PB7:     .ASCIZ  /%N%ADUP Message Type: %06/
1372 011477  PB8:     .ASCIZ  /%N%ADUP message number: %06/
1373 011533  PB9:     .ASCIZ  /%N%AMSCP Controller model # : %03/
1374 011575  PB10:    .ASCIZ  /%N%A    Microcode version # : %03/
1375 011637  PB11:    .ASCIZ  /%N%AController is IDLE when it should be ACTIVE running format program/
1376 011746  PB13:    .ASCIZ  /%N/
1377 011751  PF2:     .ASCIZ  /%N%N%AFinished local program without procedure error/
1378 012036  PBF0:    .ASCIZ  /%N%AFormat Parameter table entry at byte %06%N%Ais out of range/
1379 012136  PBF1:    .ASCIZ  /%N%AFormat Parameter table entry at byte %06%N%Ais incompatible with entry at byte %06/
1380 012265  PBF2:    .ASCIZ  /%N%AUNIT %06%A does not exist on controller/
1381 012341  PBF3:    .ASCIZ  /%N%AUNIT %06%A does exist but doesn't respond on controller/
1382 012435  PBF4:    .ASCIZ  /%N%AUNIT %06%A is write protected /
1383 012500  PBF5:    .ASCIZ  /%N%AWrite Fault detected on UNIT %06/
1384 012545  PBF6:    .ASCIZ  /%N%AAttempt to step hd %03%A at cyl %03%A failed on UNIT %06/
1385 012642  PBF7:    .ASCIZ  /%N%AAttempt to format hd %03%A at cyl %03%A failed on UNIT %06/
1386 012741  PBF8:    .ASCIZ  /%N%ATo many Bad Blocks              total Bad Blocks %06/
1387 013031  PBF9:    .ASCIZ  /%N%ADisk Controller model : %D3/
1388 013071  PBF10:   .ASCIZ  /%N%A    Microcode version : %D3/
1389 013131  PB11crn: .ASCIZ  /%N%AExpected CRN %06%A,Received CRN %06/
1390 013201  PB11op:  .ASCIZ  /%N%ACMDpkt Opcode %06%A,RSPpkt Opcode %06/
1391 013253  PB11sts: .ASCIZ  /%N%AResponse pkt status %06/
1392 013307  PB11end: .ASCIZ  /%N%ANo end bit(200) in response packet endcode/
1393 013366  PB11GDS: .ASCIZ  /%N%AGet Dust Status cmd/
1394 013416  PB11ESP: .ASCIZ  /%N%AExecute Supplied Prg cmd/
```

SEQ 0032

DISK PARAMETER QUESTIONS

```
1395 013453  PB11ELP: .ASCIZ  /%N%AExecute Local Prg cmd/
1396 013505  PB11SD: .ASCIZ  /%N%ASend Data cmd/
1397 013527  PB11RD: .ASCIZ  /%N%AReceive Data cmd/
1398 013554  PB11AP: .ASCIZ  /%N%AAbort Prg cmd/
1399 013576  pb11s0: .ASCIZ  /%N%Asts:  successful/
1400 013623  pb11s1: .ASCIZ  /%N%Asts:  Invalid Command/
1401 013655  pb11s2: .ASCIZ  /%N%Asts:  No Region Available/
1402 013713  pb11s3: .ASCIZ  /%N%Asts:  No Region Suitable/
1403 013750  pb11s4: .ASCIZ  /%N%Asts:  Program Not Known/
1404 014004  pb11s5: .ASCIZ  /%N%Asts:  Load Failure/
1405 014033  pb11s6: .ASCIZ  /%N%Asts:  Standalone/
1406 014060  pb11s9: .ASCIZ  /%N%Asts:  Host Buffer Access error/
1407 014123  pb1201: .ASCIZ  /%N%ASA er: Envelope\packet Read (parity or timeout)/
1408 014207  pb1202: .ASCIZ  /%N%ASA er: Envelope\packet Write (parity or timeout)/
1409 014274  pb1203: .ASCIZ  /%N%ASA er: Controller ROM and RAM parity/
1410 014345  pb1204: .ASCIZ  /%N%ASA er: Controller RAM parity/
1411 014406  pb1205: .ASCIZ  /%N%ASA er: Controller ROM parity/
1412 014447  pb1206: .ASCIZ  /%N%ASA er: Queue Read (parity or timeout)/
1413 014521  pb1207: .ASCIZ  /%N%ASA er: Queue Write (parity or timeout)/
1414 014574  pb1208: .ASCIZ  /%N%ASA er: Interrupt Master/
1415 014630  pb1209: .ASCIZ  /%N%ASA er: Host Access Timeout (higher level protocol dependent)/
1416 014731  pb1210: .ASCIZ  /%N%ASA er: Credit Limit Exceeded /
1417 014773  pb1211: .ASCIZ  /%N%ASA er: Bus Master Error/
1418 015027  pb1212: .ASCIZ  /%N%ASA er: Diagnostic Controller Fatal error/
1419 015104  pb1213: .ASCIZ  /%N%ASA er: Instruction Loop Timeout/
1420 015150  pb1214: .ASCIZ  /%N%ASA er: Invalid Connection Identifier/
1421 015221  pb1215: .ASCIZ  /%N%ASA er: Interrupt Write Error/
1422 015262  pb1216: .ASCIZ  /%N%ASA er: MAINTENANCE READ\WRITE Invalid Region Identifier/
1423 015356  pb1217: .ASCIZ  /%N%ASA er: MAINTENANCE WRITE Load to non-loadable controller/
1424 015453  pb1218: .ASCIZ  /%N%ASA er: Controller RAM error (non-parity)/
1425 015530  pb1219: .ASCIZ  /%N%ASA er: INIT sequence error/
1426 015567  pb1220: .ASCIZ  /%N%ASA er: High level protocol incompatibility error/
1427 015654  pb1221: .ASCIZ  /%N%ASA er: Purge\poll hardware failure/
1428 015723  pb1222: .ASCIZ  /%N%ASA er: Mapping Register read error (parity or timeout)/
1429 016016  pb1223: .ASCIZ  /%N%ASA er: Attempt to set port data transfer mapping when option not present/
1430 016133  PB12:   .ASCIZ  /%N%ASA Value (oct) %06/
1431
1432 016162  PBsf0:  .ASCIZ  /%N%ADUP type %06%A message number %06/
1433 016230  DRPunt: .ASCIZ  /%N%ATEST UNIT %06%A was dropped/
1434 016270  TYPASC: .ASCIZ  /%N%APLEASE TYPE ANSWER to controller question or just <return>/
1435          ;
```

```
FORMAT Messages
   1437                    .sbttl  FORMAT Messages
   1438
   1439            ; queries
   1440
   1441 016367  qfuit:  ;.byte  2.,.b.spl                ; Unit Info Table? (spl #2)
   1442 016367           .asciz  '%N%AEnter UIT:'
   1443 016406  qfdat:  ; byte  0.,.a.que                ; Date? (que #0)
   1444 016406           .asciz  'Enter date <MM-DD-YYYY>:'
   1445 016437  dfunt:  ;.byte  1.,.a.def                ; Unit? (def #1)
   1446 016437           .asciz  'Enter unit number to format <0>:'
   1447 016500  dfbad:  ;.byte  4.,.a.def                ; Use Bad? (def #4)
   1448 016500           .asciz  'Use existing bad block information <N>:'
   1449 016550  dfdwn:  ;.byte  5.,.a.def                ; Downline? (def #5)
   1450 016550           .asciz  'Use down-line load <N>:'
   1451 016600  dfcon:  ;.byte  6.,.a.def                ; Continue? (def #6)
   1452 016600           .asciz  'Continue if bad block information is inaccessible <N>:'
   1453 016667  qfser:  ; byte  7.,.a.que                ; Serial #? (que #7)
   1454 016667           .asciz  'Enter non-zero serial number <9 digits>:'
   1455
   1456            ; Informational Messages
   1457
   1458 016740  sfbegt: ;.byte  0.,.a.inf                ; Begin (inf #0)
   1459 016740           .asciz  '%N%AFormat Begun'
   1460 016761  sfdont: ;.byte  1.,.a.inf                ; Complete (inf #1)
   1461 016761           .asciz  '%N%AFormat complete'
   1462 017005  sfrevt: ;.byte  2.,.a.inf                ; # of Revectored LBNs (inf #2)
   1463 017005           .asciz  '% Revectored LBNS'
   1464 017027  sfr1t:  ;.byte  3.,.a.inf                ; # of primary ... (inf #3)
   1465 017027           .asciz  '% Primary revectored LBNS'
   1466 017061  sfr2t:  ;.byte  4.,.a.inf                ; # of secondary ... (inf #4)
   1467 017061           .asciz  '% Secondary/tertiary revectored LBNS'
   1468 017126  sfrcbt: ;.byte  5.,.a.inf                ; # of Bad RCT blocks ... (inf #5)
   1469 017126           .asciz  '% Bad blocks in the RCT area due to data errors'
   1470 017206  sfdbbt: ;.byte  7.,.a.inf                ; # of Bad DBNs ... (inf #7)
   1471 017206           .asciz  '% Bad blocks in the DBN area due to data errors'
   1472 017266  sfxbbt: ;.byte  9.,.a.inf                ; # of Bad XBNs ... (inf #9)
   1473 017266           .asciz  '% Bad blocks in the XBN area due to data errors'
   1474         017346  sftryt: ;.byte  11.,.a.inf                ; # of Retries (inf #11)
   1476 017411  sfrbbt: ;.byte  14.,.a.inf                ; # of Bad RBNs ... (inf #14)
   1477 017411           .asciz  '% Bad RBNS'
   1478 017424  sfcylt: ;.byte  15.,.a.inf                ; Formatting Cyl (inf #15)
   1479 017424           .asciz  'Formatting Cyl %'
```

SEQ 0034

```
FORMAT Messages
   1481          ; Successful Termination Messages
   1482
   1483              ;.byte      12.,.a.ter              ; Reformat Worked (ter #12)
   1484 017445  sffcut: .asciz   '%N%AFCT used successfully'
   1485
   1486              ;.byte      13.,.a.ter              ; Reconstruct Worked (ter #13)
   1487 017477  sffcnt: .asciz   '%N%AFCT was not used'
   1488
   1489          ; Error messages
   1490
   1491 017524  efstat: ;.byte   1.,.a.fat              ; Status Error (fat #1)
   1492 017524          .asciz   '%N%AGET STATUS failure'
   1493
   1494 017553  efsndt: ;.byte   2.,.a.fat              ; Send Error (fat #2)
   1495 017553          .asciz   '%N%AQ-PORT send error'
   1496
   1497 017601  efcmdt: ;.byte   3.,.a.fat              ; Command Error (fat #3)
   1498 017601          .asciz   '%N%AUnsuccessful command'
   1499
   1500 017632  efrcvt: ;.byte   4.,.a.fat              ; Receive Error (fat #4)
   1501 017632          .asciz   '%N%AQ PORT receive error'
   1502
   1503 017663  efbusf: ;.byte   5.,.a.fat              ; Bus Error (fat #5)
   1504 017663          .asciz   '%N%AQ-Bus I/O error'
   1505
   1506 017707  efinit: ;.byte   6.,.a.fat              ; Format Init Error (fat #6)
   1507 017707          .asciz   '%N%AFormatter initialization error'
   1508
   1509 017752  efnut:  ;.byte   7.,.a.fat              ; Unit nonexistent error (fat #7)
   1510 017752          .asciz   '%N%ANonexistent unit number'
   1511
   1512 020006  efdxft: ;.byte   8.,.a.fat              ; DBN/XBN Format error (fat #8)
   1513 020006          .asciz   '%N%ADBN/XBN format error (drive FORMAT command failed)'
   1514
   1515 020075  effcct: ;.byte   9.,.a.fat              ; FCT copies error (fat #9)
   1516 020075          .asciz   '%N%AFCT does not have enough good copies of each block
   1517
   1518 020164  efsekt: ;.byte   10.,.a.fat             ; Seek error (fat #10)
   1519 020164          .asciz   '%N%ASEEK error'
   1520
   1521 020203  efrcct: ;.byte   11.,.a.fat             ; RCT copies error (fat #11)
   1522 020203          .asciz   '%N%ARCT does not have enough good copies of each block'
   1523
   1524 020272  eflbft: ;.byte   12.,.a.fat             ; LBN format error (fat #12)
   1525 020272          .asciz   '%N%ALBN format error (drive FORMAT command failed)'
   1526
   1527 020355  effcwt: ;.byte   13.,.a.fat             ; FCT write error (fat #13)
   1528 020355          .asciz   '%N%AFCT write error (check write protect switch)'
   1529
   1530 020436  efrcrt: ;.byte   14.,.a.fat             ; RCT read error (fat #14)
   1531 020436          .asciz   '%N%ARCT read error'
   1532
   1533 020461  efrcwt: ;.byte   15.,.a.fat             ; RCT write error (fat #15)
   1534 020461          .asciz   '%N%ARCT write error'
   1535
   1536 020505  efrcft: ;.byte   16.,.a.fat             ; RCT full error (fat #16)
   1537 020505          .asciz   '%N%ARCT full'
```

```
FORMAT Messages
   1538
   1539 020522  effcrt: ;.byte  17.,.a.fat              ; FCT read error (fat #17)
   1540 020522           .asciz  '%N%AFCT read error'
   1541
   1542 020545  effcnt: ;.byte  18.,.a.fat              ; FCT nonexistent error (fat #18)
   1543 020545           .asciz  '%N%AFCT nonexistent'
   1544
   1545 020571  effcdt: ;.byte  19.,.a.fat              ; FCT downline load error (fat #19)
   1546 020571           .asciz  '%N%AFCT Down-line load error'
   1547
   1548 020626  eftmot: ;.byte  20 ,.a.fat              ; Drive timeout error (fat #20)
   1549 020626           .asciz  '%N%ADrive init timeout'
   1550
   1551 020655  efillt: ;.byte  21.,.a.fat              ; Illegal response error (fat #21)
   1552 020655           .asciz  '%N%AIllegal response to start-up question'
   1553
   1554 020727  efwart: ;.byte  22.,.a fat              ; Head error (fat #22)
   1555 020727           .asciz  '%N%AWARNING - possible head addressing problem - run diagnostics'
   1556
   1557 021030  efinpt: ;.byte  23.,.a.fat              ; Input error (fat #23)
   1558 021030           .asciz  '%N%AINPUT Error '
   1559
   1560 021051  efmedt: ;.byte  24., a.fat              ; Media error (fat #24)
   1561 021051           .asciz  '%N%AMedia degraded'
   1562                                 .list   bin
   1563                                  .EVEN
```

```
                                .sbttl global subroutines
                                ;*****************************************************************
                                ;
                                ;       THIS ROUTINE WILL ALTER THE CPU INTERRUPT LEVEL TO THREE,
                                ;       INITIATE THE MUT POLLING AND WAIT A REASONABLE AMOUNT OF
                                ;       TIME HOPING THAT THE MUT WILL INTERRUPT.
                                ;There is really 2 sections to this routine. One section awaits the
                                ;normal course of a command by waiting for an iterupt and then going
                                ;The other course of the routine is to handled timeout commands. This is
                                ;
                                ;a little more difficult. If there is a command that is time-out the
                                ;program will do a GETDUST status command to find out the status. IF
                                ;we get an interrupt there are 2 possibilties.
                                ; 1. a response to the Get Dust status command. Handled like any other
                                ;command.
                                ; 2. a response to the timeout command in which case we handle it instead
                                 of the normal handler located in the program right after it was issued.
                                ;This means that there is a GET DUST response pending which we must handle
                                ;shortly after. As soon as we handle the GET DUST response we check to
                                ; make sure we handled the timed out command by checking the LSTCRN register
                                ; and return to the DUP dialog mode by checking the DUP message type and
                                ;responding to the intial timed out command
                                ;
                                ;  }>>> waiting for initial or timed out command
                                ;  }    }interrupt received
                                ;  }    }just jump to vector address location
                                ;  }    }or on site response handler located right after cmd loader in the program
                                ;  }    }
                                ;  }    }if no interrupt then save cmd info & submit GETDUST command
                                ;}>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>waiting for GETDUST status & timed out command responses, if
 no interrupt then fatal err
                                ;} }                            }interrupt received
                                ;} }                            }test for response to GET DUST or TIMED_OUT cmd
                                ;} }                            }handle response here
                                ;} }                            }if response was GET DUST check if we handle the TIMED_OUT c
md already
                                ;} }    }<<<<<<<<<<<<<<<<<<<<<<<}(yes, LSTCRN   ^) go back to DUP dialog mode by checking DU
P type and responding
                                ;} }<<<<<<<<<<<<<<<<<<<<<<<<<<<<}(no,  LSTCRN = 0) load TIMED_OUT cmd Vector,Opcode,CRN goto
 top of routine
                                ;}<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<}if response was TIMED_OUT cmd clear LSTCRN and wait for GET
 DUST response
                                ;       }
                                ;       } go to DUP dialog routine
                                ;*****************************************************************
     021074                     POLLWT:
     021074  106427  000140             mtps    #140                    ;DROP CPU LEVEL TO THREE.
     021100  062706  000004             add     #4,sp                   ;fix stack for interrupt
                                                                        ;(2 for subroutine )
     021104  005777  161150             tst     @IPreg                  ;TELL MUT TO START POLLING.

     021110  012700  000015             mov     #15,r0          ;outer wait loop
     021114  012701  000140     1$:     mov     #140,r1         ;       ;
     021120  004737  024450     2$:     jsr     pc,BIT15T       ;       ;inner wait loop
     021124                             BREAK                   ;       ;
     021126  077104                     sob     r1,2$           ;       ;
     021130  077007                     sob     r0,1$           ;
                                                                ;No response has been received
     021132  004737  024450             jsr     pc,BIT15T       ;test SA make sure not a fatal error
     021136  013700  002402             mov     cmdpak+10,r0    ;get opcode
     021142  022700  000001             cmp     #op.gds,r0      ;if the command issued was a GETDUST STATUS and time
out big trouble
     021146  001006                     bne     GDS0            ;if not go do a GET DUST to find out what the situat
ion is
```

```
   1622 021150                                          ERRDF   12,df14                 ;type no interrupt after get dust status command cont
roller dead
   1623 021160  000137  032376                         jmp     dropunt                 ;drop unit and go on
   1624
   1625                                 ;GETDUST                                        ;save timed out command information
   1626 021164  017737  161072  002464  GDS0:   mov     @vector,LSTVCT          ;store the vector address of timeout command
   1627 021172  013737  002372  002460          mov     cmdpak,LSTCRN           ;store the CRN of the timed out command
   1628 021200  013737  002402  002462          mov     cmdpak+10,LSTCMD        ;store the opcode of timed out command
   1629
   1630 021206  032737  100000  002370          bit     #bit15,cmdlen+2         ;test ownership of ring make sure we own it
   1631 021214  001363                          bne     GDS0                    ;if we don t own it wait until we do
   1632 021216  012737  000016  002366          mov     #14.,cmdlen             ;load lenght of packet to be send
   1633 021224  112737  000000  002370          movb    #0,cmdlen+2             ;load msg type and credit
   1634 021232  112737  000002  002371          movb    #dup.id,cmdlen+3        ;load DUP connection ID
   1635 021240  005237  002372                  inc     cmdpak                  ;load new CRN
   1636 021244  005037  002374                  clr     cmdpak+2
   1637 021250  005037  002376                  clr     cmdpak+4
   1638 021254  005037  002400                  clr     cmdpak+6
   1639 021260  012737  000001  002402          mov     #op.gds,cmdpak+10       ;load up opcode
   1640 021266  005037  002404                  clr     cmdpak+12               ;no modifiers
   1641
   1642 021272  012777  021334  160762          mov     #RFDO,@vector           ;NEW VECTOR PLACE
   1643 021300  012737  002272  002446          mov     #rsppak,rsprng          ;load response packet area into ring
   1644 021306  012737  002372  002452          mov     #cmdpak,cmdrng          ;load command packet area into ring
   1645 021314  012737  140000  002450          mov     #140000,RSPRNG+2        ;PORT OWNERSHIP BIT.
   1646 021322  012737  100000  002454          mov     #bit15,CMDRNG+2
   1647 021330  000137  021074                  jmp     POLLWT                  ;GO and wait for interrupt
   1648
   1649
   1650
   1651                         ;*****************************************************************
   1652 021334                     RFDO:                                       ;INTR TO HERE if GETDUST or TIMED_OUT cmd
   1653                         ;*
   1654                         ; There is only 3 ways out code.
   1655                         ;If GETDUST resposne and TIMED_OUT cmd response was handled
   1656                         ; if LSTCRN = 0 and RSPPAK+10 = OP.GDS+OP.END then
   1657                         ; back to DUP dialog mode.
   1658                         ;or
   1659                         ; (TIMED_OUT cmd still hasn't returned but GETDUST has returned)
   1660                         ; if LSTCRN = 0 and RSPPAK+10 = OP.GDS+OP.END then
   1661                         ; check if idle or active. if idle then error
   1662                         ; check for progress in progress indicator if no progress then error
   1663                         ; load LSTVCT into @vector,LSTCRN into cmdpak, LSTCMD into cmdpak+10
   1664                         ; set response ring ownership to Port Owned
   1665                         ; jmp to pollwt.
   1666                         ;or
   1667                         ; (TIMED_OUT cmd response received before GETDUST response returned)
   1668                         ; if LSTCRN = 0 and RSPPAK+10 not= OP.GDS+OP.END then
   1669                         ; clear LSTCRN and
   1670                         ; jmp to pollwt.
   1671
   1672 021334  013701  002372                  mov     cmdpak,r1               ;check command packet CRN
   1673 021340  013700  002272                  mov     rsppak,r0               ;check response packet CRN
   1674 021344  020001                          cmp     r0,r1                   ;Are they the SAME must be GETDUST cmd
   1675 021346  001101                          bne     3$                      ; if not it must be the TIMED_OUT cmd
   1676
   1677 021350  023727  002302  000201          cmp     rsppak+10,#op.gds+op.end        ;it should be a GETDUST lets make sure
   1678 021356  001412                          beq     1$
```

```
Global subroutines

   1679 021360                                          printf  @pb10                  ;unexpected cmd response in t me out loop
   1680 021400  000137  032362                          jmp     unkwn                  ;error handler
   1681
   1682 021404  004737  023516          1$:             jsr     pc,RSPCHK              ;check the response
   1683 021410  005737  002460                          tst     LSTCRN                 ;see if timed out command was already recieved (lstc
rn <> 0)
   1684 021414  001002                                  bne     2$
   1685 021416  000137  027514                          jmp     DUPDLG                 ;if Timed out cmd was already received then goto DUP
 dialog mode
   1686
   1687 021422                          2$:                                            ;if Timed out command was not received already (LSTC
RN not= 0)
   1688 021422  132737  000010  002311                  bitb    @bit3,rsppak+17        ;if server idle then error
   1689 021430  001010                                  bne     1002$                  ;if not check for progress
   1690 021432                                          printf  @pb11                  ;controller idle when it should be active
   1691
   1692 021452  013700  002312          1002$:          mov     rsppak+20,r0           ;check for progress in progress indicator
   1693 021456  013701  002314                          mov     rsppak+22,r1
   1694 021462  020037  002466                          cmp     r0,loprgi              ;see if low word of progress indicator is the same a
s older value
   1695 021466  001007                                  bne     1001$                  ;if it is then continue
   1696 021470  020137  002470                          cmp     r1,hiprgi              ;see if high vaule is the same
   1697 021474  001004                                  bne     1001$
   1698 021476                                          ERRDF   11,DF13                ;no progress shown after cmd timeout
   1699
   1700 021506  010037  002466          1001$:          mov     r0,loprgi              ;update progress indicator
   1701 021512  010137  002470                          mov     r1,hiprgi
   1702 021516  013737  002460  002372                  mov     LSTCRN,cmdpak          ;move TIMED_OUT cmd CRN into cmd
   1703 021524  013737  002462  002402                  mov     LSTCMD,cmdpak+10       ;move TIMED_OUT cmd Opcode into cmd
   1704 021532  013777  002464  160522                  mov     LSTVCT,@vector         ;load TIMED_OUT cmd interrupt handler address into v
ector
   1705 021540  012737  140000  002450                  mov     #140000,RSPRNG+2       ;Port owned
   1706 021546  000137  021074                          jmp     pollwt                 ;wait for TIMED_OUT cmd response
   1707
   1708
   1709
   1710 021552  020037  002460          3$:             cmp     r0,LSTCRN              ;check the crn with the last CRN from the timeout co
mmand
   1711 021556  001412                                  beq     4$                     ;
   1712 021560                                          printf  @pb10                  ;Unexpected cmd response in time out loop
   1713 021600  000137  032362                          jmp     unkwn                  ;error handler
   1714
   1715                                                                                ;Timed out command recieved but Get Dust Status is s
till in Queue
   1716 021604  013737  002460  002372  4$:             mov     LSTCRN,cmdpak          ;load timed out command values for RSPCHK routine
   1717 021612  013737  002462  002402                  mov     LSTCMD,cmdpak+10       ;load timed out command values for RSPCHK routine
   1718 021620  005037  002460                          clr     LSTCRN                 ;if it is the timeout command clear LAST CRN registe
r
   1719 021624  004737  023516                          jsr     pc,RSPCHK              ;go check the command
   1720 021630  000137  021074                          jmp     POLLWT                 ;go wait for GETDUST intterupt
   1721
   1722
   1723                                 ;***************************************************************
   1724                                 ;
   1725                                 ;       This routine builds the UIT table or get the UIT table
   1726                                 ; depending who the questions are answered to the manual questions.
   1727                                 ; If the unit is a listed or regconizable drive we will use a prebuilt
   1728                                 ; UIT table. If not we will have to ask all the questions to build
   1729                                 ; a table.
   1730                                 ;
   1731                                 ;***************************************************************
   1732 021634                          BLDUIT:
   1733 021634  000240                          nop
   1734 021636                                  printf  @DrvTxa                        ;print out UIT tables and their related drives
   1735 021656                                  printf  @DrvTxb                        ;_IN    Drive
```

Global subroutines

```
   1736 021676                                   printf  #DrvTx0               ;0      rd51
   1737 021716                                   printf  #DrvTx1               ;1      rd52
   1738 021736                                   printf  #DrvTx2               ;2      etc
   1739 021756                                   printf  #DrvTx3               ;3      etc
   1740 021776                                   printf  #DrvTx4               ;4
   1741 022016                                   printf  #DrvTx5
   1742 022036                                   printf  #DrvTx6
   1743 022056                                   printf  #DrvTx7
   1744 022076                                   printf  #DrvTxc
   1745
   1746
   1747 022116                           GMANID unt.nbr,UIN,0,17,0,10,no        ;GET Unit identifier number (0-7)
   1748                                                                         ;PLACE IN bits 0 3.
   1749                                                                         ;no defaults person must know what Unit Identificati
on number.
   1750 022136  013702  002264                   mov     UIN,r2
   1751 022142  032702  0000[illegible]0                   bit     #UITothr,r2              ;see if they choose other
   1752 022146  001012                           bne     tblbld                  ;if no UIT present then build a UIT in default UIT d
ata area
   1753
   1754 022150  012703  003000                   mov     #UITC,r3                ;r3 contains base address of UIT tables
   1755 022154  001403                           beq     2$                      ;if UIN=0 then set table to UIT0
   1756 022156  062703  000144           1$:     add     #UITsiz,r3              ;else multiply UIT size by the UIN number and add to
 base address
   1757 022162  077203                           sob     r2,1$
   1758
   1759 022164  010337  002256           2$:     mov     r3,UITadr               ;store the proper address of the UIT table
   1760 022170  000137  023402                   jmp     cont                    ;continue initialization
   1761
   1762 022174                           tblbld:                                 ;We must build a UNIT INFORMATION TABLE
   1763 022174  012737  004440  002256           mov     #UITdf,UITadr           ;move the address of UIT table into a register
   1764 022202                           GMANID TBQ0,UITdf+0,0,-1,0,-1,yes               ;
   1765 022222                           GMANID TBQ1,UITdf+2,0,-1,0,-1,yes               ;
   1766 022242                           GMANID TBQ2,UITdf+4,0,-1,0,-1,yes               ;
   1767 022262                           GMANID TBQ3,UITdf+6,0,-1,0,-1,yes               ;
   1768 022302                           GMANID TBQ4,UITdf+10,0,-1,0,-1,yes              ;
   1769 022322                           GMANID TBQ5,UITdf+12,0,-1,0,-1,yes              ;
   1770 022342                           GMANID TBQ6,UITdf+14,0,-1,0,-1,yes              ;
   1771 022362                           GMANID TBQ7,UITdf+16,0,-1,0,-1,yes              ;
   1772 022402                           GMANID TBQ8,UITdf+20,0,-1,0,-1,yes              ;
   1773 022422                           GMANID TBQ9,UITdf+22,0,-1,0,-1,yes              ;
   1774 022442                           GMANID TBQ10,UITdf+24,0,-1,0,-1,yes             ;
   1775 022462                           GMANID TBQ11,UITdf+26,0,-1,0,-1,yes             ;
   1776 022502                           GMANID TBQ12,UITdf+30,0,-1,0,-1,yes             ;
   1777 022522                           GMANID TBQ13,UITdf+32,0,-1,0,-1,yes             ;
   1778 022542                           GMANID TBQ14,UITdf+34,0,-1,0,-1,yes             ;
   1779 022562                           GMANID TBQ15,UITdf+36,0,-1,0,-1,yes             ;
   1780 022602                           GMANID TBQ16,UITdf+40,0,-1,0,-1,yes             ;
   1781 022622                           GMANID TBQ17,UITdf+42,0,-1,0,-1,yes             ;
   1782 022642                           GMANID TBQ18,UITdf+44,0,-1,0,-1,yes             ;
   1783 022662                           GMANID TBQ19,UITdf+46,0,-1,0,-1,yes             ;
   1784 022702                           GMANID TBQ20,UITdf+50,0,-1,0,-1,yes             ;
   1785 022722                           GMANID TBQ21,UITdf+52,0,-1,0,-1,yes             ;
   1786 022742                           GMANID TBQ22,UITdf+54,0,-1,0,-1,yes             ;
   1787 022762                           GMANID TBQ23,UITdf+56,0,-1,0,-1,yes             ;
   1788 023002                           GMANID TBQ24,UITdf+60,0,-1,0,-1,yes             ;
   1789 023022                           GMANID TBQ25,UITdf+62,0,-1,0,-1,yes             ;
   1790 023042                           GMANID TBQ26,UITdf+64,0,-1,0,-1,yes
   1791                                  ;tbq27 purpose [illegible]
   1792 023062                           GMANID [illegible] yes
```

Global subroutines

```
                                GMANID TBQ29,UITdf+70,0,-1,0,-1,yes             ;
                                GMANID TBQ30,UITdf+72,0,-1,0,-1,yes             ;
                                GMANID TBQ31,UITdf+74,0,-1,0,-1,yes             ;
                                GMANID TBQ32,UITdf+76,0,-1,0,-1,yes             ;
                                GMANID TBQ33,UITdf+100,0,-1,0,-1,yes            ;
                                GMANID TBQ34,UITdf+102,0,-1,0,-1,yes            ;
                                GMANID TBQ35,UITdf+104,0,-1,0,-1,yes            ;
                                GMANID TBQ36,UITdf+106,0,-1,0,-1,yes            ;
                                GMANID TBQ37,UITdf+110,0,-1,0,-1,yes            ;
                                GMANID TBQ38,UITdf+112,0,-1,0,-1,yes            ;
                                GMANID TBQ39,UITdf+114,0,-1,0,-1,yes            ;
                                GMANID TBQ40,UITdf+116,0,-1,0,-1,yes            ;

023402  000207                  cont:   rts     pc                      ;go back
                                ;*****************************************************************
                                ;
                                ; This routine types out the ASCII information passed
                                ; by the disk controller. This ASCII information is
                                ; contained in the buffer called DATARE and is offset
                                ; by 1 word. To fake the DRS macro routine a "%A" is
                                ; placed in front of the text.
                                ;*****************************************************************

023404                          typDUPbuf:
023404  012701  002472                  mov     #datare,r1      ;get data area address of ascii info
023410  063701  002306                  add     rsppak+14,r1    ;add the number of byte transfered
023414  105021          1$:     clrb    (r1)+           ;put null characters into data buffer after end of ASCII inf
o
023416  020127  002616                  cmp     r1,#prgnam      ;
023422  001374                  bne     1$              ;we do this to fake out the DRS macro

023424  112737  000045  002472        movb    #45,datare      ;put the "%" delimiter for the DRS macro
023432  112737  000101  002473        movb    #101,datare+1   ;put the 'A" for ascii info for the DRS macro
023440                          printx  #PB13           ;New Line <cr><lf>
023460                          printx  #datare         ;print the message returned from the controller

023500                          clrDUPbuf:
023500  012701  002472                  mov     #datare,r1      ;clear out entire data area
023504  105021          2$:     clrb    (r1)+           ;
023506  020127  002616                  cmp     r1,#prgnam      ;
023512  001374                  bne     2$              ;
023514  000207                  rts     pc
                                ;*****************************************************************
                                ;
                                ;       THIS ROUTINE IS TO CHECK ON THE RESPONSE PACKET
                                ;       GOODNESS. THE COMMAND REFERENCE NUMBER, THE END CODE
                                ;       AND THE STATUS ARE TESTED.
                                ;*****************************************************************

023516                          RSPCHK:

023516  013701  002372              mov         cmdpak,r1
023522  013700  002272              mov         rsppak,r0
023526  020001                      cmp         r0,r1                   ;compare CRN numbers
023530  001014                      bne         1$
023532  013701  002402              mov         cmdpak+10,r1
023536  062701  000200              add         #200,r1
023542  013700  002302              mov         rsppak+10,r0
```

Global subroutines

```
1850 023546  020001                    cmp     r0,r1                   ;compare Opcodes
1851 023550  001004                    bne     1$
1852 023552  013701  002304            mov     rsppak+12,r1            ;check the status
1853 023556  001001                    bne     1$
1854 023560  000207                    rts     pc                      ;if all checks then return
1855
1856                                                                   ;if all doesn't check then a bad packet
1857 023562                    1$:     ERRDF   10,df11                 ;Bad response packet
1858 023572                    PRNTpkt:
1859 023572                            Printb  #PB11crn,cmdpak,rsppak  ;Expected CRN XXXX ,Received CRN YYYY
1860 023622  013701  002302            mov     rsppak+10,r1            ;check response opcode reply
1861 023626  032701  000200            bit     #200,r1                 ;see if a end command response was send
1862 023632  001010                    bne     2$
1863 023634                            printx  #PB11end                ;No end bit in response packet endcode
1864 023654  022701  000201    2$:     cmp     #201,r1
1865 023660  001010                    bne     3$                      ;check if Get Dust Status command
1866 023662                            printx  #PB11GDS
1867 023702  022701  000202    3$:     cmp     #202,r1
1868 023706  001010                    bne     4$                      ;check if Execute Supplied Program
1869 023710                            printx  #PB11ESP
1870 023730  022701  000203    4$:     cmp     #203,r1
1871 023734  001010                    bne     5$                      ;check if Execute Local Program
1872 023736                            printx  #PB11ELP
1873 023756  022701  000204    5$:     cmp     #204,r1
1874 023762  001010                    bne     6$                      ;check if Send Data
1875 023764                            printx  #PB11SD
1876 024004  022701  000205    6$:     cmp     #205,r1
1877 024010  001022                    bne     7$                      ;check if Receive Data
1878 024012                            printx  #PB11RD
1879 024032                            Printb  #PBSF0,r3,r5    ;"type xxx, message number xxxxx is unknow to this program"
1880 024056  022701  000206    7$:     cmp     #206,r1
1881 024062  001010                    bne     8$                      .check if Abort Program
1882 024064                            printx  #PB11AP
1883 024104                    8$:     Printb  #PB11op,cmdpak+10,rsppak+10
1884                                                                   ;CMDpkt opcode XXXX,RSPpkt opcode YYYY
1885
1886 024134  013701  002304            mov     rsppak+12,r1            ;find out what kind of status we have
1887 024140  022701  000000            cmp     #0.,r1
1888 024144  001010                    bne     10$
1889 024146                            printx  #pb11s0                 ;status   successful
1890 024166  022701  000001    10$:    cmp     #1.,r1
1891 024172  001010                    bne     11$
1892 024174                            printx  #pb11s1                 ;status:  Invalid Command
1893 024214  022701  000002    11$:    cmp     #2.,r1
1894 024220  001010                    bne     12$
1895 024222                            printx  #pb11s2                 ;status:  No Region Available
1896 024242  022701  000003    12$:    cmp     #3.,r1
1897 024246  001010                    bne     13$
1898 024250                            printx  #pb11s3                 ;status:  No Region Suitable
1899 024270  022701  000004    13$:    cmp     #4.,r1
1900 024274  001010                    bne     14$
1901 024276                            printx  #pb11s4                 ;status:  Program Not Known
1902 024316  022701  000005    14$:    cmp     #5.,r1
1903 024322  001010                    bne     15$
1904 024324                            printx  #pb11s5                 ;status:  Loac Failure
1905 024344  022701  000006    15$:    cmp     #6.,r1
1906 024350  001010                    bne     16$
```

Global subroutines

```
1907 024352                                     printx  #pb11s6                ;status:  Standalone
1908 024372  022701  000011          16$:      cmp     #9.,r1
1909 024376  001010                             bne     19$
1910 024400                                     printx  #pb11s9                ;status·  Host Buffer Access error
1911 024420                          19$:
1912 024420                                     Printb  #PB11sts,rsppak+12     ;Response packet status XXYX
1913 024444  000137  032376                     jmp     dropunt                ;drop unit and go on
1914
1915
1916                                 ;*************************************************************************
1917                                 ;
1918                                 ;                          BIT FIFTEEN TEST
1919                                 ;*************************************************************************
1920 024450                          BIT15T:
1921 024450  032714  100000                     bit     #bit15,(r4)
1922 024454  001001                             bne     100$
1923 024456  000207                             rts     pc
1924 024460                          100$:      ERRDF   9,df12          ;Fatal SA error
1925 024470  011401                             mov     (r4),r1
1926 024472  022701  001000                     cmp     #1000,r1
1927 024476  001010                             bne     1$
1928 024500                                     printx  #pb1201         ;
1929 024520  022701  100001          1$:        cmp     #100001,r1
1930 024524  001010                             bne     2$
1931 024526                                     printx  #pb1202         ;
1932 024546  022701  100002          2$:        cmp     #100002,r1
1933 024552  001010                             bne     3$
1934 024554                                     printx  #pb1203         ;
1935 024574  022701  100003          3$:        cmp     #100003,r1
1936 024600  001010                             bne     4$
1937 024602                                     printx  #pb1204         ;
1938 024622  022701  100004          4$:        cmp     #100004,r1
1939 024626  001010                             bne     5$
1940 024630                                     printx  #pb1205         ;
1941 024650  022701  100005          5$:        cmp     #100005,r1
1942 024654  001010                             bne     6$
1943 024656                                     printx  #pb1206         ;
1944 024676  022701  100006          6$:        cmp     #100006,r1
1945 024702  001010                             bne     7$
1946 024704                                     printx  #pb1207         ;
1947 024724  022701  100007          7$:        cmp     #100007,r1
1948 024730  001010                             bne     8$
1949 024732                                     printx  #pb1208         ;
1950 024752  022701  100010          8$:        cmp     #100010,r1
1951 024756  001010                             bne     9$
1952 024760                                     printx  #pb1209         ;
1953 025000  022701  100011          9$:        cmp     #100011,r1
1954 025004  001010                             bne     10$
1955 025006                                     printx  #pb1210         ;
1956 025026  022701  100012          10$:       cmp     #100012,r1
1957 025032  001010                             bne     11$
1958 025034                                     printx  #pb1211         ;
1959 025054  022701  100013          11$:       cmp     #100013,r1
1960 025060  001010                             bne     12$
1961 025062                                     printx  #pb1212         ;
1962 025102  022701  100014          12$:       cmp     #100014,r1
1963 025106  001010                             bne     13$
```

Global subroutines

```
1964 025110                                     printx  #pb1213         ;
1965 025130  022701  100015          13$:       cmp     #100015,r1
1966 025134  001010                             bne     14$
1967 025136                                     printx  #pb1214         ;
1968 025156  022701  100016          14$:       cmp     #100016,r1
1969 025162  001010                             bne     15$
1970 025164                                     printx  #pb1215         ;
1971 025204  022701  100017          15$:       cmp     #100017,r1
1972 025210  001010                             bne     16$
1973 025212                                     printx  #pb1216         ;
1974 025232  022701  100020          16$:       cmp     #100020,r1
1975 025236  001010                             bne     17$
1976 025240                                     printx  #pb1217         ;
1977 025260  022701  100021          17$:       cmp     #100021,r1
1978 025264  001010                             bne     18$
1979 025266                                     printx  #pb1218         ;
1980 025306  022701  100022          18$:       cmp     #100022,r1
1981 025312  001010                             bne     19$
1982 025314                                     printx  #pb1219         ;
1983 025334  022701  100023          19$:       cmp     #100023,r1
1984 025340  001010                             bne     20$
1985 025342                                     printx  #pb1220         ;
1986 025362  022701  100024          20$:       cmp     #100024,r1
1987 025366  001010                             bne     21$
1988 025370                                     printx  #pb1221         ;
1989 025410  022701  100025          21$:       cmp     #100025,r1
1990 025414  001010                             bne     22$
1991 025416                                     printx  #pb1222         ;
1992 025436  022701  100026          22$:       cmp     #100026,r1
1993 025442  001010                             bne     23$
1994 025444                                     printx  #pb1223         ;
1995 025464                          23$:
1996 025464                                     printb  #pb12,r1        ;SA value: xxxxx
1997 025506  000137  032376                     jmp     dropunt                 ;drop unit and go on
1998
1999                                 ;*************************************************************************
2000                                 ;       Unexpected Interrupt Server
2001                                 ;
2002                                 ;*************************************************************************
2003 025512                          intsrv:
2004
2005 025512                                     ERRSF   8,sf100 ;Fatal SA error
2006 025522                                     docln                   ;do clean up and quit
2007 025524  000137  032376                     jmp     dropunt         ;drop test unit and end pass
2008
2009
```

Global subroutines

```
                                BGNPROT
                                .WORD -1
                                .WORD -1
                                .WORD -1
                                ENDPROT

                                 BGNINIT                        ;SEQUENTIAL EXAMPLE
                                 READEF         #EF.CONTINUE    ;Continue COMMAND?
                                 BCOMPLETE      conton          ;YES, GET NO P-TABLE but still initialize
                                 READEF         #EF.NEW         ;NEW PASS
                                 BNCOMPLETE     next            ;if not new then go to next unit number
                                 MANUAL                         ;check if in manual mode if not exit test
                                 BCOMPLETE      setup           ;program has to have manual intervention
                                      jmp       abort
SETUP:                           mov       #-1,LOGLNIT         ;INITIALIZE LOGICAL UNIT NBR
NEXT:                            inc       LOGUNIT                     ;POINT TO NEXT LOGICAL UNIT
                                 cmp       LOGUNIT,L$UNIT      ;HAVE WE PASSED MAXIMUM?
                                 bne       1$
                                 jmp       ABORT               ;YES, ABORT THE PASS
1$:                              GPHARD LOGUNIT,PLOC           ;GET THE P-TABLE
                                 BNCOMPLETE NEXT               ;if not available get next unit

                                 mov       ploc,r0
                                 mov       r0,ptbl                     ;store the Ptable address for unit
conton:                          mov       (r0)+,ipreg                 ;store IPreg address into register
                                 mov       (r0)+,vector                ;store vector
                                 mov       (r0),UIN                    ;store the UIN value

                                 clr       LSTCRN                      ;basic initialization stuff
                                 clr       LSTVCT
                                 clr       LOPRGI
                                 clr       HIPRGI

                                 mov       @#4,-(sp)                   ;test to see if controller is there
                                 mov       $2,@#4
                                 clr       @IPreg                      ;get controller into know state
                                 br        $3

$2:                              ERRDF     7,DF4                       ;NXM trap at controller IP address
                                 dodu      LOGUNIT                     ;drop unit
                                 br        next                        ;get new unit

$3:                              mov       (sp)+,@#4                   ;move value back into location 4

                                 mov       #76,r0              ;clean out all packets and interrupt flags
                                 mov       #rsp1,r1            ;and the command area
$4:                              clr       (r1)+
                                 sob       r0,$4

                                 br        end

ABORT:                           DOCLN                         ;DO CLEAN-UP AND ABORT THE PASS
END:                             ENDINIT                       ;FINISHED


                                BGNAUTO
                                DODU LOGUNIT
```

Global subroutines

```
   2068 025770                          ENDAUTO
   2069
   2070 025772                          BGNCLN
   2071 025772  005077  154262                clr     @IPreg          ;get controller into know state
   2072 025776                                Break                   ;waste some time
   2073 026000                          ENDCLN
   2074
   2075 026002                          BGNDU
   2075 026002                                printf #DRPunt,LOGUNIT
   2077 026026                          ENDDU
   2078
```

```
   2080 026030                           BGNTST 1
   2081                                 ;************************************************************************
   2082                                 ;
   2083                                 ; This format starts by initializating
   2084                                 ;       r1 = ptable address
   2085                                 ;       r3 = step bit mask
   2086                                 ;       r4 = SA address
   2087                                 ;************************************************************************
   2088 026030                          HRDINT:
   2089 026030  106427  000140                  mtps    #140
   2090 026034  013704  002260                  mov     ipreg,r4
   2091 026040  012703  004000                  mov     #bit11,r3       ;initialization step reg
   2092 026044  005024                  1$:     clr     (r4)+           ; start initialaztion IP and put SA adrs into r4
   2093 026046                                  Break
   2094
   2095 026050  012700  040000                  mov     #40000,r0       ;looking for step 1 >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
>>>>>>>>>>>>>>>>>
   2096 026054  004737  024450          4$:     jsr     pc,bit15T       ;Look for fatal SAreg error
   2097 026060  030314                          bit     r3,(r4)         ;check sa reg for step 1       bit11 =0 >1
   2098 026062  001002                          bne     6$
   2099 026064  077005                          sob     r0,4$
   2100 026066  000562                          br      timout
   2101
   2102 026070  013700  002262          6$:     MOV     vector,r0       ;responding in step 1 <<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
<<<<<<<<<<<<<<<<<
   2103 026074  000241                          clc
   2104 026076  006200                          asr     r0
   2105 026100  006200                          asr     r0
   2106 026102  052700  100200                  bis     #<bit15+bit7>,r0 ;= VECTOR/4
   2107 026106  013701  002262                  mov     vector,r1
   2108 026112  012721  026142                  mov     #sp2int,(r1)+   ;load up interrupt loacation into vector
   2109 026116  012711  000140                  mov     #140,(r1)       ;lower the priority
   2110 026122  010014                          mov     r0,(r4)         ; Enable INTERRUPTS,set 1 cmd rng & 1 rsp rng
   2111
   2112 026124  006303                          asl     r3              ;next step
   2113 026126  012700  070000                  mov     #70000,r0       ;looking for step 2 >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
>>>>>>>>>>>>>>>>>
   2114 026132                          8$:     break                   ;waste some time
   2115 026134                                  break                   ;waste some time
   2116 026136  077003                          sob     r0 8$           ;wait ten seconds
   2117 026140  000535                          br      timout          ;timed out if here
   2118
   2119 026142                          sp2int:
   2120 026142  004737  024450                  jsr     pc,bit15T       ;look for sa error
   2121 026146  030314                          bit     r3,(r4)         ;check step
   2122 026150  001551                          beq     wrngstep
   2123
   2124 026152  012700  002446          12$:    mov     #RSPRNG,R0      ;responding in step 2 <<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
<<<<<<<<<<<<<<<<<
   2125 026156  042700  000001                  bic     #bit0,r0        ;no adapter purge interrupts
   2126 026162  013701  002262                  mov     vector,r1
   2127 026166  012721  026216                  mov     #sp3int,(r1)+   ;load up interrupt loacation into vector
   2128 026172  012711  000140                  mov     #140,(r1)       ;lower the priority
   2129 026176  010014                          mov     r0,(r4)         ;load low ringbase address of the communications area
   2130
   2131 026200  006303                          asl     r3              ;next step
   2132 026202  012700  070000                  mov     #70000,r0       ;looking for step 3 >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
>>>>>>>>>>>>>>>>>
   2133 026206                          13$:    break                   ;waste some time
   2134 026210                                  break                   ;waste some time
   2135 026212  077003                          sob     r0,13$          ;wait ten seconds
   2136 026214  000507                          br      timout
```

```
   2137
   2138 026216  004737  024450          sp3int: jsr     pc,bit15T
   2139 026222  030314                          bit     r3,(r4)
   2140 026224  001523                          beq     wrngstep
   2141
   2142 026226  013701  002262                  mov     vector,r1       ;responding in step 3 <<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
<<<<<<<<<<<<<<<<
   2143 026232  012721  026262                  mov     #sp4int,(r1)+   ;load up interrupt loacation into vector
   2144 026236  012711  000140                  mov     #140,(r1)       ;lower the priority
   2145 026242  005014                          clr     (r4)            ;load low ringbase address of the communications area
   2146
   2147 026244  006303                          asl     r3
   2148 026246  012701  070000                  mov     #70000,r1       ;looking for step 4 >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
>>>>>>>>>>>>>>>>>
   2149 026252                          18$:    break                   ;waste some time
   2150 026254                                  break                   ;waste some time
   2151 026256  077 _3                          sob     r1,18$          ;wait ten seconds
   2152 026260  000465                          br      timout
   2153
   2154 026262  004737  024450          sp4int: jsr     pc,bit15T       ;check for SA error
   2155 026266  030314                          bit     r3,(r4)         ;check for step 4
   2156 026270  001501                          beq     wrngstep
   2157
   2158 026272  011401                          mov     (r4),r1         ;identify the controller number and microcode version number
   2159 026274  010100                          mov     r1,r0
   2160 026276  006201                          asr     r1              ;shift version # out
   2161 026300  006201                          asr     r1
   2162 026302  006201                          asr     r1
   2163 026304  006201                          asr     r1
   2164 026306  042701  177400                  bic     #177400,r1      ;clear top bits off
   2165 026312                                  printb  #pb9,R1         ;print the disk controller model number
   2166 026334  042700  177760                  bic     #177760,r0      ;clear model number out
   2167 026340                                  printb  #pb10,R0        ;print microcode version number in decimal
   2168
   2169 026362  122701  000007                  cmpb    #Mrqdx1,r1      ;check for Model #
   2170 026366  001462                          beq     gobit
   2171 026370  122701  000023                  cmpb    #Mrqdx3,r1      ;check for Model #
   2172 026374  001457                          beq     gobit
   2173
   2174 026376                                  ERRSOFT 6,SFT1          ; DEVICE FATAL wrong model #,wrong controller
   2175 026406                                  printf  #pb12,r1                ; Model # listed XXXXX
   2176 026430  000137  026534                  jmp     gobit                   ;drop unit and go on
   2177
   2178 026434                          timout: ERRDF   5,DF1           ; DEVICE FATAL controller timeout during hard init
   2179 026444                                  Printf  #pb1,r3,(r4)    ; Expected SA step bit xxxxx set, received yyyyyy
   2180 026470  000137  032376                  jmp     dropunt                 ;drop unit and go on
   2181
   2182 026474                          wrngstep:
   2183 026474                                  ERRDF   4,DF2           ; DEVICE FATAL wrong step bit set after interrupt
   2184 026504                                  Printf  #pb1,r3,(r4)    ; Expected SA step bit xxxxx, received in SA yyyyyy
   2185 026530  000137  032376                  jmp     dropunt                 ;drop unit and go on
   2186 026534                          GOBIT:
   2187 026534  012714  000001                  mov     #1,(r4)                 ;Controller is NOW INITIALIZED
   2188
   2189 026540                          GETDUST                         ;Do a Get Dust Status command start things off
        026540  032737  100000  002370  GDS2:   bit     #bit15,cmdlen+2         ;test ownership of ring make sure we own it
        026546  001374                          bne     GDS2                    ;if we con't own it wait until we do
        026550  012737  000016  002366          mov     #14.,cmdlen             ;load lenght of packet to be send
        026556  112737  000000  002370          movb    #0,cmdlen+2             ;load msg type and credit
```

Global subroutines

```
        026564  112737  000002  002371          movb    #dup.id,cmdlen+3        ;load DUP connection ID
        026572  005237  002372                  inc     cmdpak                  ;load new CRN
        026576  005037  002374                  clr     cmdpak+2
        026602  005037  002376                  clr     cmdpak+4
        026606  005037  002400                  clr     cmdpak+6
        026612  012737  000001  002402          mov     #op.gds,cmdpak+10       ;load up opcode
        026620  005037  002404                  clr     cmdpak+12               ;no modifiers

        026624  012777  026666  153430          mov     #RFD2,@vector           ;NEW VECTOR PLACE
        026632  012737  002272  002446          mov     #rsppak,rsprng          ;load response packet area into ring
        026640  012737  002372  002452          mov     #cmdpak,cmdrng          ;load command packet area into ring
        026646  012737  140000  002450          mov     #140000,RSPRNG+2        ;PORT OWNERSHIP BIT.
        026654  012737  100000  002454          mov     #bit15,CMDRNG+2
        026662  004737  021074                  jsr     pc,POLLWT               ;GO TO POLL AND WAIT ROUTINE.
                                        ;*******************************************************************
        026666                          RFD2:                                   ;INTR TO HERE.
        026666  012777  025512  153366          mov     #intsrv,@vector         ;CHANGE VECTOR
        026674  004737  023516                  jsr     pc,RSPCHK
                                                                                ;GO TO ROUTINE THAT WILL CHECK ON
                                                                                ;THE RESPONSE RECVD FROM THE MUT.
                                                                                ;IT WILL CHECK THE CMD REF
                                                                                ;NUM, THE ENDCODE AND STATUS.
   2190 026700  132737  000010  002311          bitb    #bit3,rsppak+17         ;is this server active already
   2191 026706  001465                          beq     ELPcmd                  ;branch to Execute Local Program
   2192 026710                                  ERRSOFT 3,SFTO                  ;Soft Error "already active  will do an ABORT cmd"
   2193 026720                          ABRT                                    ;Doing an ABRT do get into idle state
        026720  032737  100000  002370  ABRT3:  bit     #bit15,cmdlen+2         ;test ownership of ring make sure we own it
        026726  001374                          bne     ABRT3                   ;if we don't own it wait until we do
        026730  012737  000016  002366          mov     #14.,cmdlen             ;load lenght of packet to be send
        026736  112737  000000  002370          movb    #0,cmdlen+2             ;load msg type and credit
        026744  112737  000002  002371          movb    #dup.id,cmdlen+3        ;load DUP connection ID
        026752  005237  002372                  inc     cmdpak                  ;load new CRN
        026756  005037  002374                  clr     cmdpak+2
        026762  005037  002376                  clr     cmdpak+4
        026766  005037  002400                  clr     cmdpak+6
        026772  012737  000006  002402          mov     #op.abrt,cmdpak+10      ;load up opcode
        027000  005037  002404                  clr     cmdpak+12               ;no modifiers

        027004  012777  027046  153250          mov     #RFD3,@vector           ;NEW VECTOR PLACE
        027012  012737  002272  002446          mov     #rsppak,rsprng          ;load response packet area into ring
        027020  012737  002372  002452          mov     #cmdpak,cmdrng          ;load command packet area into ring
        027026  012737  140000  002450          mov     #140000,RSPRNG+2        ;PORT OWNERSHIP BIT.
        027034  012737  100000  002454          mov     #bit15,CMDRNG+2
        027042  004737  021074                  jsr     pc,POLLWT               ;GO TO POLL AND WAIT ROUTINE.
                                        ;*******************************************************************
        027046                          RFD3:                                   ;INTR TO HERE.
        027046  012777  025512  153206          mov     #intsrv,@vector         ;CHANGE VECTOR
        027054  004737  023516                  jsr     pc,RSPCHK
                                                                                ;GO TO ROUTINE THAT WILL CHECK ON
                                                                                ;THE RESPONSE RECVD FROM THE MUT.
                                                                                ;IT WILL CHECK THE CMD REF
                                                                                ;NUM, THE ENDCODE AND STATUS.
   2194 027060  000627                          br      GDS2                    ;branch back to make sure not busy
   2195
   2196 027062                          ELPcmd:
   2197 027062                          GMANID ASK.prg,PRGnam,A,-1,6.,6.,yes    ;ask for the User what local program he wants to run
   2198
```

Global subroutines

```
   2199 027102                                  EXLCPRG PRGnam                          ;Execute Local program "FORMAT" or what ever they wr
ote
        027102  032737  100000  002370  ELP4:   bit     #bit15,cmdlen+2                 ;test ownership of ring make sure we own it
        027110  001374                          bne     ELP4                            ;if we don't own it wait until we do
        027112  012737  000022  002366          mov     #22,cmdlen                      ;load lenght of packet to be send
        027120  112737  000000  002370          movb    #0,cmdlen+2                     ;load msg type and credit
        027126  112737  000002  002371          movb    #dup.id,cmdlen+3                ;load DUP connection ID
        027134  005237  002372                  inc     cmdpak                          ;load new CRN
        027140  005037  002374                  clr     cmdpak+2
        027144  005037  002376                  clr     cmdpak+4
        027150  005037  002400                  clr     cmdpak+6
        027154  012737  000003  002402          mov     #op.elp,cmdpak+10               ;load up opcode
        027162  012737  000001  002404          mov     #stdaln,cmdpak+12               ;stand alone modifier
        027170  012700  000006                  mov     #6,r0                           ;6 letters transfer
        027174  012701  002406                  mov     #cmdpak+14,r1                   ;starting address to place program name
        027200  012702  002616                  mov     #PRGnam,r2                      ;start of Program Name
        027204  112221                  rfdj4:  movb    (r2)+,(r1)+                     ;add 2 to bycnt then store
        027206  077002                          sob     r0,rfdj4

        027210  012777  027252  153044          mov     #RFD4,@vector                   ;NEW VECTOR PLACE
        027216  012737  002272  002446          mov     #rsppak,rsprng                  ;load response packet area into ring
        027224  012737  002372  002452          mov     #cmdpak,cmdrng                  ;load command packet area into ring
        027232  012737  140000  002450          mov     #140000,RSPRNG+2                ;PORT OWNERSHIP BIT.
        027240  012737  100000  002454          mov     #bit15,CMDRNG+2
        027246  004737  021074                  jsr     pc,POLLWT                       ;GO TO POLL AND WAIT ROUTINE.
                                                ;*****************************************************************
        027252                                      RFD4:                               ;INTR TO HERE.
        027252  012777  025512  153002              mov     #intsrv,@vector             ;CHANGE VECTOR
        027260  004737  023516                      jsr     pc,RSPCHK
                                                                                        ;GO TO ROUTINE THAT WILL CHECK ON
                                                                                        ;THE RESPONSE RECVD FROM THE MUT.
                                                                                        ;IT WILL CHECK THE CMD REF
                                                                                        ;NUM, THE ENDCODE AND STATUS.
   2200
   2201 027264  122737  000011  002311          cmpb    #bit3+bit0,rsppak+17            ;is this program a standalone,DUP dialog type
   2202 027272  001406                          beq     1$
   2203 027274                                  ERRDF   2,DF3                           ;"Device Fatal can't do remote programs"
   2204 027304  000137  032376                  jmp     dropunt                         ;drop unit and go on
   2205 027310                              1$:
   2206 027310                              RECVDAT #datare,#80.
        027310  032737  100000  002370  RCD5:   bit     #bit15,cmdlen+2                 ;test ownership of ring make sure we own it
        027316  001374                          bne     RCD5                            ;if we don't own it wait until we do
        027320  012737  000034  002366          mov     #34,cmdlen                      ;load lenght of packet to be send
        027326  112737  000000  002370          movb    #0,cmdlen+1                     ;load msg type and credit
        027334  112737  000002  002371          movb    #dup.id,cmdlen+3                ;load DUP connection ID
        027342  005237  002372                  inc     cmdpak                          ;load new CRN
        027346  005037  002374                  clr     cmdpak+2
        027352  005037  002376                  clr     cmdpak+4
        027356  005037  002400                  clr     cmdpak+6
        027362  012737  000005  002402          mov     #op.rec,cmdpak+10               ;load up opcode
        027370  005037  002404                  clr     cmdpak+12                       ;no modifiers
        027374  012737  000120  002406          mov     #80.,cmdpak+14
        027402  005037  002410                  clr     cmdpak+16
        027406  012737  002472  002412          mov     #datare,cmdpak+20               ;load address of buffer describtor
        027414  005037  002414                  clr     cmdpak+22
        027420  005037  002416                  clr     cmdpak+24
        027424  005037  002420                  clr     cmdpak+26
        027430  005037  002422                  clr     cmdpak+30
```

Global subroutines

```
        027434  005037  002424                  clr     cmdpak+32

        027440  012777  027502  152614          mov     #RFD5,@vector           ;NEW VECTOR PLACE
        027446  012737  002272  002446          mov     #rsppak,rsprng          ;load response packet area into ring
        027454  012737  002372  002452          mov     #cmdpak,cmdrng          ;load command packet area into ring
        027462  012737  140000  002450          mov     #140000,RSPRNG+2        ;PORT OWNERSHIP BIT.
        027470  012737  100000  002454          mov     #bit15,CMDRNG+2
        027476  004737  021074                  jsr     pc,POLLWT               ;GO TO POLL AND WAIT ROUTINE.
                                        ;*****************************************************************
        027502                          RFD5:                                   ;INTR TO HERE.
        027502  012777  025512  152552          mov     #intsrv,@vector         ;CHANGE VECTOR
        027510  004737  023516                  jsr     pc,RSPCHK
                                                                                ;GO TO ROUTINE THAT WILL CHECK ON
                                                                                ;THE RESPONSE RECVD FROM THE MUT.
                                                                                ;IT WILL CHECK THE CMD REF
                                                                                ;NUM, THE ENDCODE AND STATUS.
   2207                                 ;*
   2208                                 ;       get
   2209                                 ;       r3 = type
   2210                                 ;       r4 = SA adrs
   2211                                 ;       r5 = sub number
   2212                                 ;-
   2213 027514  113703  002473          DUPDLG: movb    datare+1,r3     ;get dup type info
   2214 027520  006203                          asr     r3
   2215 027522  006203                          asr     r3
   2216 027524  006203                          asr     r3
   2217 027526  006203                          asr     r3
   2218 027530  042703  177760                  bic     #type,r3        ;mask off all but DUP type
   2219                                 ;       printx  #PB7,r3         ;"received DUP command type XX'
   2220 027534  013705  002472                  mov     datare,r5       ;get dup message number info
   2221 027540  042705  170000                  bic     #msgnbr,r5      ;clear out top 4 bits
   2222                                 ;       printx  #PB8,r5         ;"received dup message number XX"
   2223
   2224
   2225                                 ;
   2226                                 ; Check for the type.
   2227                                 ; If QUESTION type, it will be answered by sending
   2228                                 ; an answer through a Send command which will be followed
   2229                                 ; by a Receive command to await further instructions.
   2230                                 ;
   2231                                 ; If a DEFAULT QUESTION type is given an answer will
   2232                                 ; either be given or a blank send command returned.
   2233                                 ; Either way we will do a Send command followed by a
   2234                                 ; Receive command.
   2235                                 ;
   2236                                 ; If INFORMATIONAL type, check message number and type
   2237                                 ; information according to message number given.
   2238                                 ;
   2239                                 ; If FATAL ERROR type, check message number and print
   2240                                 ; error message accordingly. No other commands will
   2241                                 ; be given following this type of command.
   2242                                 ;
   2243                                 ; If TERMINATION type check the message number and print the
   2244                                 ; correct message. Ussually this implies a succesful
   2245                                 ; end to the formatter. After this command we exit the program
   2246                                 ;
   2247                                 ; If SPECIAL type we are asking for the FCT table to be passed
```

Global subroutines

```
                                        ; to the RQDX3 controller. We will send the table with a Send
                                        ; command and then to a Receive command to proceed.
                                        ;
2251 027544  022703  000001     qstn:   cmp     #Question,r3    ;test for "question" subtype
2252 027550  001054                     bne     dfqstn          ;if not branch
2253 027552  122737  000106  002616     cmpb    @'F,prgnam      ;if running the format program then print info
2254 027560  001034                     bne     qnbra           ;else just go for an answer
2255
2256 027562  004737  023500     qnbr0:  jsr     pc,clrDUPbuf    ;clear out data buffer so DRS macros don't show default
2257 027566  022705  000000             cmp     @0,r5           ;check for message number
2258 027572  001012                     bne     qnbr7           ;check for next message number
2259 027574                             GMANID  qfdat,DATARE,A,177777,10  10.,no                    ;DATE MM-DD-YYYY ?
2260 027614  000137  030124             jmp     SDT6            ;branch to Send Data command
2261
2262 027620  022705  000007     qnbr7:  cmp     @7,r5           ;check for message number
2263 027624  001012                     bne     qnbra           ;check for next message number
2264 027626                             GMANID  qfser,DATARE,A,177777,9.,9.,NO                    ;SERIAL NUMBER 9 digits ?
2265 027646  000137  030124             jmp     SDT6
2266
2267 027652  004737  023404     qnbra:  jsr     pc,typDUPbuf    ;type out ASCII sent by disk controller
2268 027656                             GMANID  ASK.ANSWER,DATARE,A,177777,0.,10.,YES             ;give it an answer
2269 027676  000137  030124             jmp     SDT6            ;branch to Send Data command
2270
2271
2272
2273
2274 027702  022703  000002     dfqstn: cmp     @DefQuest,r3    ;test for 'Default Question  subtype
2275 027706  001406                     beq     dqnbr1
2276 027710  000137  030334             jmp     infrm           ;if not branch
2277 027714  122737  000106  002616     cmpb    @'F,prgnam      ;if running the format program then print info
2278 027722  001066                     bne     dqnbra          ;else just go for an answer
2279
2280 027724  004737  023500     dqnbr1: jsr     pc,clrDUPbuf    ;clear out data buffer so DRS macros don t show default
2281 027730  022705  000001             cmp     @1,r5           ;check for message number
2282 027734  001012                     bne     dqnbr4          ;check for next message number
2283 027736                             GMANID  dfunt,DATARE,A,177777,0,3,YES                     ;UNIT NUMBER 0 255 ?
2284 027756  000137  030124             jmp     SDT6            ;branch to Send Data command
2285
2286 027762  022705  000004     dqnbr4: cmp     @4,r5           ;check for message number
2287 027766  001012                     bne     dqnbr5          ;check for next message number
2288 027770                             GMANID  dfbad,DATARE,A,177777,0,1,YES                     ;Use existing bad block 'nfo
rmation (Y or N)?
2289 030010  000137  030124             jmp     SDT6            ;branch to Send Data command
2290
2291 030014  022705  000005     dqnbr5. cmp     @5,r5           ;check for message number
2292 030020  001012                     bne     dqnbr6          ;check for next message number
2293 030022                             GMANID  dfdwn,DATARE,A,177777,0,1,YES                     ;Use Down Line Load (Y or N)
?
2294 030042  000137  030124             jmp     SDT6            ;branch to Send Data command
2295
2296 030046  022705  000006     dqnbr6: cmp     @6,r5           ;check for message number
2297 030052  001012                     bne     dqnbra          ;check for next message number
2298 030054                             GMAN'   dfcon,DATARE,A,177777,0,1,YES                     ;Continue  f bad block 'nfor
mation is inaccessable (Y or N)?
2299 030074  000137  030124             jmp     SDT6
2300
2301                                                            ;if unknown use default and continue
2302                                                            ;who knows maybe 't will be useful some day
2303 030100  004737  023404     dqnbra: jsr     pc,typDUPbuf    ;type out ASCII sent by d'sk controller
2304 030104                             GMANID  ASK.ANSWER,DATARE,A,177777,0.,10.,YES             ;g'.e 't an answer
```

Global subroutines

```
2305
2306 030124                                        SENDDAT @datare,@10.    ;sent the answer                "STD6"
     030124  032737  100000  002370  SDT6:   bit     @bit15,cmdlen+2         ;test ownership of ring make sure we own it
     030132  001374                          bne     SDT6                    ;if we don't own it wait until we do
     030134  012737  000034  002366          mov     @34,cmdlen              ;load lenght of packet to be send
     030142  112737  000000  002370          movb    @0,cmdlen+2             ;load msg type and credit
     030150  112737  000002  002371          movb    @dup.id,cmdlen+3        ;load DUP connection ID
     030156  005237  002372                  inc     cmdpak                  ;load new CRN
     030162  005037  002374                  clr     cmdpak+2
     030166  005037  002376                  clr     cmdpak+4
     030 72  005037  002400                  clr     cmdpak+6
     030176  012737  000004  002402          mov     @op.sen,cmdpak+10       ;load up opcode
     030204  005037  002404                  clr     cmdpak+12               ;no modifiers
     030210  012737  000012  002406          mov     @10.,cmdpak+14
     030216  005037  002410                  clr     cmdpak+16
     030222  012737  002472  002412          mov     @datare,cmdpak+20       ,load address of buffer describtor
     030230  005037  002414                  clr     cmdpak+22
     030234  005037  002416                  clr     cmdpak+24
     030240  005037  002420                  clr     cmdpak+26
     030244  005037  002422                  clr     cmdpak+30
     030250  005037  002424                  clr     cmdpak+32

     030254  012777  030316  152000          mov     @RFD6,@vector           ;NEW VECTOR PLACE
     030262  012737  002272  002446          mov     @rsppak,rsprng          ;load response packet area into ring
     030270  012737  002372  002452          mov     @cmdpak,cmdrng          ;load command packet area into ring
     030276  012737  140000  002450          mov     @140000,RSPRNG+2        ;PORT OWNERSHIP BIT.
     030304  012737  100000  002454          mov     @bit15,CMDRNG+2
     030312  004737  021074                  jsr     pc,POLLWT               ;GO TO POLL AND WAIT ROUTINE.
                                        ;*************************************************************************
     030316                              RFD6:                               ;INTR TO HERE.
     030316  012777  025512  151736          mov     @intsrv,@vector         ;CHANGE VECTOR
     030324  004737  023516                  jsr     pc,RSPCHK
                                                                             ;GO TO ROUTINE THAT WILL CHECK ON
                                                                             ;THE RESPONSE RECVD FROM THE MUT.
                                                                             ;IT WILL CHECK THE CMD REF
                                                                             ;NUM, THE ENDCODE AND STATUS.
2307 030330  000137  027310                  jmp     RCD5            ;do another receive cmd
2308
2309
2310
2311 030334  022703  000003          infrm:  cmp     @Inform,r3      ;test for "Informational" subtype
2312 030340  001042                          bne     term            ;if not branch
2313 030342  122737  000106  002616          cmpb    @'F,prgnam      ;if running the format program then print info
2314 030350  001032                          bne     inbra
2315
2316 030352  022705  000000          inbr0:  cmp     @0,r5           ;check for message number
2317 030356  001012                          bne     inbr1           ;check for next message number
2318 030360  004737  023500                  jsr     pc,clrDUPbuf    ;clear out DUP buffer so there is no echo on last ASCII
2319 030364                                  printf  @sfbegt         ;format begun
2320 030404  022705  000001          inbr1:  cmp     @1,r5           ;check for message number
2321 030410  001012                          bne     inbra           ;check for next message number
2322 030412  004737  023500                  jsr     pc,clrDUPbuf    ;clear out DUP buffer so there is no echo on last ASCII
2323 030416                                  printf  @sfdont         ;format complete
2324
2325 030436  004737  023404          inbra:  jsr     pc,typDUPbuf    ;type out ASCII sent by disk controller
2326 030442  000137  027310                  jmp     PCD5            ;do another receive command
2327
```

Global subroutines

```
2328
2329
2330 030446  022703  000004          term:   cmp     #terminat,r3    ;test for termination type
2331 030452  001052                          bne     ftler           ;if not branch
2332 030454  122737  000106  002616          cmpb    #'F,prgnam      ;if running the format program then branch to error routine
2333 030462  001032                          bne     tnbra
2334
2335 030464  022705  000014          tnbr12: cmp     #12.,r5         ;test for sub number #1
2336 030470  001012                          bne     tnbr13          ;branch if not sub number #1
2337 030472                                  printf  #sffcut
2338 030512  000137  032376                  jmp     dropunt         ;drop test unit and end pass
2339
2340 030516  022705  000015          tnbr13: cmp     #13.,r5         ;test for msg number
2341 030522  001012                          bne     tnbra           ;branch if not right number
2342 030524                                  printf  #sffcnt
2343 030544  000137  032376                  jmp     dropunt         ;drop test unit and end pass
2344
2345 030550  004737  023404          tnbra:  jsr     pc,typDUPbuf    ;type out ASCII sent by disk controller
2346 030554                                  printf  #PF2            ;print finished local program without procedure error
2347 030574  000137  032404                  jmp     etst            ;end DUP diaglog but stay in test loop
2348
2349
2350
2351 030600  022703  000005          ftler:  cmp     #Ftlerr,r3      ;test for "Fatal Error" subtype
2352 030604  001402                          beq     1$
2353 030606  000137  032052                  jmp     spcl            ;if not branch
2354 030612  122737  000106  002616  1$:     cmpb    #'F,prgnam      ;if running the format program then branch to error routine
2355 030620  001414                          beq     2$
2356 030622  004737  023404                  jsr     pc,typDUPbuf    ;type out ASCII sent by disk controller
2357 030626                                  printf  #DF15           ;Fatal error reported when running local program
2358 030646  000137  032376                  jmp     dropunt         ;drop unit and end pass
2359
2360 030652                          2$:     ERRHRD  1,HRDO          ;Hard device  error
2361
2362 030662  022705  000001          fnbr1:  cmp     #1,r5           ;test for sub number #1
2363 030666  001012                          bne     fnbr2           ;branch if not sub number #1
2364 030670                                  printb  #efstat         ;"GET STATUS failure"
2365 030710  000137  032404                  jmp     etst            ;end DUP diaglog but stay in test loop
2366
2367 030714  022705  000002          fnbr2:  cmp     #2.,r5          ;test for msg number
2368 030720  001012                          bne     fnbr3           ;branch if not right number
2369 030722                                  printf  #efsndt         ;
2370 030742  000137  032404                  jmp     etst            ;end DUP diaglog but stay in test loop
2371
2372 030746  022705  000003          fnbr3:  cmp     #3.,r5          ;test for msg number
2373 030752  001012                          bne     fnbr4           ;branch if not right number
2374 030754                                  printf  #efcmdt         ;
2375 030774  000137  032404                  jmp     etst            ;end DUP diaglog but stay in test loop
2376
2377 031000  022705  000004          fnbr4:  cmp     #4.,r5          ;test for msg number
2378 031004  001012                          bne     fnbr5           ;branch if not right number
2379 031006                                  printf  #efrcvt         ;
2380 031026  000137  032404                  jmp     etst            ;end DUP diaglog but stay in test loop
2381
2382 031032  022705  000005          fnbr5:  cmp     #5.,r5          ;test for msg number
2383 031036  001012                          bne     fnbr6           ;branch if not right number
2384 031040                                  printf  #efbust         ;
```

Global subroutines

```
2385 031060  000137  032404                jmp     etst            ;end DUP diaglog but stay in test loop
2386
2387 031064  022705  000006        fnbr6:  cmp     #6.,r5          ;test for msg number
2388 031070  001012                        bne     fnbr7           ;branch if not right number
2389 031072                                printf  #efinit         ;
2390 031112  000137  032404                jmp     etst            ;end DUP diaglog but stay in test loop
2391
2392 031116  022705  000007        fnbr7:  cmp     #7.,r5          ;test for msg number
2393 031122  001012                        bne     fnbr8           ;branch if not right number
2394 031124                                printf  #efnut          ;"Q-PORT send error "
2395 031144  000137  032404                jmp     etst            ;end DUP diaglog but stay in test loop
2396
2397 031150  022705  000010        fnbr8:  cmp     #8.,r5          ;test for msg number
2398 031154  001012                        bne     fnbr9           ;branch if not right number
2399 031156                                printf  #efdxft
2400 031176  000137  032404                jmp     etst            ;end DUP diaglog but stay in test loop
2401
2402 031202  022705  000011        fnbr9:  cmp     #9.,r5          ;test for msg number
2403 031206  001012                        bne     fnbr10          ;branch if not right number
2404 031210                                printf  #effcct         ;"Q-PORT send error "
2405 031230  000137  032404                jmp     etst            ;end DUP diaglog but stay in test loop
2406
2407 031234  022705  000012        fnbr10: cmp     #10.,r5         ;test for msg number
2408 031240  001012                        bne     fnbr11          ;branch if not right number
2409 031242                                printf  #efsekt         ;"Q-PORT send error "
2410 031262  000137  032404                jmp     etst            ;end DUP diaglog but stay in test loop
2411
2412 031266  022705  000013        fnbr11: cmp     #11.,r5         ;test for msg number
2413 031272  001012                        bne     fnbr12          ;branch if not right number
2414 031274                                printf  #efrcct         ;"Q-PORT send error "
2415 031314  000137  032404                jmp     etst            ;end DUP diaglog but stay in test loop
2416
2417 031320  022705  000014        fnbr12. cmp     #12.,r5         ;test for msg number
2418 031324  001012                        bne     fnbr13          ;branch if not right number
2419 031326                                printf  #eflbft         ;"Q-PORT send error "
2420 031346  000137  032404                jmp     etst            ;end DUP diaglog but stay in test loop
2421
2422 031352  022705  000015        fnbr13: cmp     #13.,r5         ;test for msg number
2423 031356  001012                        bne     fnbr14          ;branch if not right number
2424 031360                                printf  #effcwt         ;"Q-PORT send error "
2425 031400  000137  032404                jmp     etst            ;end DUP diaglog but stay in test loop
2426
2427 031404  022705  000016        fnbr14: cmp     #14.,r5         ;test for msg number
2428 031410  001012                        bne     fnbr15          ;branch if not right number
2429 031412                                printf  #efrcrt         ;"Q PORT send error "
2430 031432  000137  032404                jmp     etst            ;end DUP diaglog but stay in test loop
2431
2432 031436  022705  000017        fnbr15: cmp     #15.,r5         ;test for msg number
2433 031442  001012                        bne     fnbr16          ;branch if not right number
2434 031444                                printf  #efrcwt         ;"Q-PORT send error "
2435 031464  000137  032404                jmp     etst            ;end DUP diaglog but stay in test loop
2436
2437 031470  022705  000020        fnbr16: cmp     #16.,r5         ;test for msg number
2438 031474  001012                        bne     fnbr17          ;branch if not right number
2439 031476                                printf  #efrcft         ;"Q-PORT send error "
2440 031516  000137  032404                jmp     etst            ;end DUP diaglog but stay in test loop
2441
```

Global subroutines

```
2442 031522  022705  000021          fnbr17: cmp     #17.,r5         ;test for msg number
2443 031526  001012                          bne     fnbr18          ;branch if not right number
2444 031530                                  printf  #effcrt         ;"Q-PORT send error "
2445 031550  000137  032404                  jmp     etst            ;end DUP diaglog but stay in test loop
2446
2447 031554  022705  000022          fnbr18: cmp     #18.,r5         ;test for msg number
2448 031560  001012                          bne     fnbr19          ;branch if not right number
2449 031562                                  printf  #effcnt         ;
2450 031602  000137  032404                  jmp     etst            ;end DUP diaglog but stay in test loop
2451
2452 031606  022705  000023          fnbr19: cmp     #19.,r5         ;test for msg number
2453 031612  001012                          bne     fnbr20          ;branch if not right number
2454 031614                                  printf  #effcdt         ;
2455 031634  000137  032404                  jmp     etst            ;end DUP diaglog but stay in test loop
2456
2457 031640  022705  000024          fnbr20: cmp     #20.,r5         ;test for msg number
2458 031644  001012                          bne     fnbr21          ;branch if not right number
2459 031646                                  printf  #eftmot         ;"Q-PORT send error "
2460 031666  000137  032404                  jmp     etst            ;end DUP diaglog but stay in test loop
2461
2462 031672  022705  000025          fnbr21: cmp     #21.,r5         ;test for msg number
2463 031676  001012                          bne     fnbr22          ;branch if not right number
2464 031700                                  printf  #efillt         ;"Q-PORT send error "
2465 031720  000137  032404                  jmp     etst            ;end DUP diaglog but stay in test loop
2466
2467 031724  022705  000026          fnbr22: cmp     #22.,r5         ;test for msg number
2468 031730  001012                          bne     fnbr23          ;branch if not right number
2469 031732                                  printf  #efwart         ;"Q-PORT send error "
2470 031752  000137  032404                  jmp     etst            ;end DUP diaglog but stay in test loop
2471
2472 031756  022705  000027          fnbr23: cmp     #23.,r5         ;test for msg number
2473 031762  000412                          br      fnbr24          ;branch if not right number
2474 031764                                  printf  #efinpt         ;"Q-PORT send error "
2475 032004  000137  032404                  jmp     etst            ;end DUP diaglog but stay in test loop
2476
2477
2478 032010  022705  000030          fnbr24: cmp     #24.,r5         ;test for msg number
2479 032014  001012                          bne     1$
2480 032016                                  printf  #efmedt
2481 032036  000137  032404                  jmp     etst            ;end DUP diaglog but stay in test loop
2482
2483 032042  004737  023404          1$:     jsr     pc,typDUPbuf    ;type out ASCII sent by disk controller
2484 032046  000137  032404                  jmp     etst            ;end DUP diaglog but stay in test loop
2485
2486
2487
2488
2489 032052  022703  000006          spcl:   cmp     #specl,r3       ;test for special type
2490 032056  001141                          bne     unkwn           ;branch if not known
2491 032060  122737  000106  002616          cmpb    #'F,prgnam      ;if running the format program then print info
2492 032066  001414                          beq     1$
2493 032070  004737  023404                  jsr     pc,typDUPbuf    ;type out ASCII sent by disk controller
2494 032074                                  printf  #DF16           ;special command issued by local program did not know how to
handle
2495 032114  000137  032362                  jmp     unkwn           ;report error
2496
2497 032120  022705  000002          1$:     cmp     #2,r5           ;test for message number 1
2498 032124  001116                          bne     unkwn           ;branch if not known
```

Global subroutines

```
2499 032126                                        printf  #qfuit           ;"enter UIT:"
2500 032146  004737  021634                        jsr     pc,bldUIT        ;go get or build UIT table
2501 032152                                        SENDDAT UITadr,#UITsiz   ;sent Unit Information table
     032152  032737  100000  002370  SDT7:       bit     #bit15,cmdlen+2          ;test ownership of ring make sure we own it
     032160  001374                              bne     SDT7                     ;if we don't own it wait until we do
     032162  012737  000034  002366              mov     #34,cmdlen               ;load lenght of packet to be send
     032170  112737  000000  002370              movb    #0,cmdlen+2              ;load msg type and credit
     032176  112737  000002  002371              movb    #dup.id,cmdlen+3         ;load DUP connection ID
     032204  005237  002372                      inc     cmdpak                   ;load new CRN
     032210  005037  002374                      clr     cmdpak+2
     032214  005037  002376                      clr     cmdpak+4
     032220  005037  002400                      clr     cmdpak+6
     032224  012737  000004  002402              mov     #op.sen,cmdpak+10        ;load up opcode
     032232  005037  002404                      clr     cmdpak+12                ;no modifiers
     032236  012737  000144  002406              mov     #UITsiz,cmdpak+14
     032244  005037  002410                      clr     cmdpak+16
     032250  013737  002256  002412              mov     UITadr,cmdpak+20         ;load address of buffer describtor
     032256  005037  002414                      clr     cmdpak+22
     032262  005037  002416                      clr     cmdpak+24
     032266  005037  002420                      clr     cmdpak+26
     032272  005037  002422                      clr     cmdpak+30
     032276  005037  002424                      clr     cmdpak+32

     032302  012777  032344  147752              mov     #RFD7,@vector            ;NEW VECTOR PLACE
     032310  012737  002272  002446              mov     #rsppak,rsprng           ;load response packet area into ring
     032316  012737  002372  002452              mov     #cmdpak,cmdrng           ;load command packet area into ring
     032324  012737  140000  002450              mov     #140000,RSPRNG+2         ;PORT OWNERSHIP BIT.
     032332  012737  100000  002454              mov     #bit15,CMDRNG+2
     032340  004737  021074                      jsr     pc,POLLWT                ;GO TO POLL AND WAIT ROUTINE.
                                                ;*************************************************************************
     032344                                         RFD7:                        ;INTR TO HERE.
     032344  012777  025512  147710                 mov     #intsrv,@vector      ;CHANGE VECTOR
     032352  004737  023516                         jsr     pc,RSPCHK
                                                                                 ;GO TO ROUTINE THAT WILL CHECK ON
                                                                                 ;THE RESPONSE RECVD FROM THE MUT.
                                                                                 ;IT WILL CHECK THE CMD REF
                                                                                 ;NUM, THE ENDCODE AND STATUS.
2502 032356  000137  027310                         jmp     RCD5         ;do another receive cmd
2503
2504
2505 032362                                 unkwn:  ERRSF   0,SF0        ; system error unkown response
2506 032372  004737  023572                         jsr     pc,PRNTpkt   ;type out packet information
2507
2508 032376                                 dropunt:
2509 032376                                         DODU    LOGUNIT      ;drop the unit
2510
2511 032404                                 etst:
2512 032404                                         docln                ;take controller offline
2513 032406                                 ENDTST
```

Global subroutines

```
   2515 032410                          BGNHRD
   2516
   2517 032412                               GPRMA ip.adr,0,0,170000,177776,YES   ;GET IP REG ADDR (170000 177776)
   2518                                                                           ;PLACE IN WORD 2 OF THE TABLE
   2519                                                                           ;DEFAULT VALUE IS FROM DEFAULT
   2520                                                                           ;TABLE.
   2521
   2522 032422                               GPRMA vec.adr,2,0,0,776,YES          ;GET THE VECTOR ADDR (OCTAL 0-776)
   2523                                                                           ;PLACE IN WORD
   2524                                                                           ;DEFAULT VALUE IS FROM DEFAULT
   2525                                                                           ;TABLE.
   2526
   2527
   2528 032432                          hrdbtm:
   2529 032432                           ENDHRD
   2530
   2531
   2532 032432                           LASTAD
        032436                          L$LAST::
   2533 032436                          BGNSETUP        1       ;number of default P-tables used 4 incase of running setup with all
units
   2534 032436                          BGNPTAB                         ;Ptable default number 0
   2535 032442  172150                         .WORD   172150                  ;IP address
   2536 032444  000154                         .WORD   154                     ;Vector address
   2537 032446  100002                         .WORD   100002                  ;Unit indentifier number rd51=1 rd52=2 rd53=3
   2538                                                                        ;bit 15 says it is from the Unit Identifier table
   2539 032450                          ENDPTAB
   2540 032450                          ENDMOD
   2541         000001                   .END
```

Symbol table

```
A      = 000000          C$DODU= 000051          DF11    010041          EF.RES= 000037 G        GDS2    026540
ABORT    025756          C$DRPT= 000024          DF12    010076          EF.STA= 000040 G        GOBIT   026534
ABRT3    026720          C$DU  = 000053          DF13    010132          ELPCMD  027062          G$CNTO= 000200
ADR    = 000020 G        C$EDIT= 000003          DF14    010206          ELP4    027102          G$DELM= 000372
ASK.AN= ****** GX        C$ERDF= 000055          DF15    010267          END     025760          G$DISP= 000003
ASK.PR   005643          C$ERHR= 000056          DF16    010357          ETST    032404          G$EXCP= 000400
ASSEMB= 000010           C$ERRO= 000060          DF2     007617          EVL    = 000004 G       G$HILI= 000002
B      = 000007          C$ERSF= 000054          DF3     007666          E$END  = 002100         G$LOLI= 000001
BIT0   = 000001 G        C$ERSO= 000057          DF4     007776          E$LOAD= 000035          G$NO   = 000000
BIT00  = 000001 G        C$ESCA= 000010          DIAGMC= 000000          FNBR1   030662          G$OFFS= 000400
BIT01  = 000002 G        C$ESEG= 000005          DQNBRA  030100          FNBR10  031234          G$OFSI= 000376
BIT02  = 000004 G        C$ESUB= 000003          DQNBR1  027724          FNBR11  031266          G$PRMA= 000001
BIT03  = 000010 G        C$ETST= 000001          DQNBR4  027762          FNBR12  031320          G$PRMD= 000002
BIT04  = 000020 G        C$EXIT= 000032          DQNBR5  030014          FNBR13  031352          G$PRML= 000000
BIT05  = 000040 G        C$FREQ= 000101          DQNBR6  030046          FNBR14  031404          G$RADA= 000140
BIT06  = 000100 G        C$FRME= 000100          DROPUN  032376          FNBR15  031436          G$RADB= 000000
BIT07  = 000200 G        C$GETB= 000026          DRPUNT  016230          FNBR16  031470          G$RADD= 000040
BIT08  = 000400 G        C$GETW= 000027          DRVTXA  004612          FNBR17  031522          G$RADL= 000120
BIT09  = 001000 G        C$GMAN= 000043          DRVTXB  004640          FNBR18  031554          G$RADO= 000020
BIT1   = 000002 G        C$GPHR= 000042          DRVTXC  005554          FNBR19  031606          G$XFER= 000004
BIT10  = 002000 G        C$GPRI= 000040          DRVTXO  004735          FNBR2   030714          G$YES  = 000010
BIT11  = 004000 G        C$INIT= 000011          DRVTX1  004756          FNBR20  031640          HIPRGI  002470
BIT12  = 010000 G        C$INLP= 000020          DRVTX2  005052          FNBR21  031672          HOE    = 100000 G
BIT13  = 020000 G        C$MANI= 000050          DRVTX3  005147          FNBR22  031724          HRDBTM  032402
BIT14  = 040000 G        C$MAP  = 000102         DRVTX4  005170          FNBR23  031756          HRDINT  026030
BIT15  = 100000 G        C$MEM  = 000031         DRVTX5  005265          FNBR24  032010          HRDO    010664
BIT15T   024450          C$MMU  = 000103         DRVTX6  005362          FNBR3   030746          IBE    = 010000 G
BIT2   = 000004 G        C$MSG  = 000023         DRVTX7  005457          FNBR4   031000          IDU    = 000040 G
BIT3   = 000010 G        C$OPNR= 000034          DUPDLG  027514          FNBR5   031032          IER    = 020000 G
BIT4   = 000020 G        C$OPNW= 000104          DUP.ID= 000002          FNBR6   031064          INBRA   030436
BIT5   = 000040 G        C$PNTB= 000014          EFBUST  017663          FNBR7   031116          INBRO   030352
BIT6   = 000100 G        C$PNTF= 000017          EFCMDT  017601          FNBR8   031150          INBR1   030404
BIT7   = 000200 G        C$PNTS= 000016          EFDXFT  020006          FNBR9   031202          INFORM= 000003
BIT8   = 000400 G        C$PNTX= 000015          EFFCCT  020075          FTLER   030600          INFRM   030334
BIT9   = 001000 G        C$PUTB= 000072          EFFCDT  020571          FTLERR= 000005          INTSRV  025512
BLDUIT   021634          C$PUTW= 000073          EFFCNT  020545          F$AU   = 000015         IPREG   002260
BOE    = 000400 G        C$QIO  = 000377         EFFCRT  020522          F$AUTO= 000020          IP.ADR  004560
CINTR    002442          C$RDBU= 000007          EFFCWT  020355          F$BGN  = 000040         ISR    = 000100 G
CLRDUP   023500          C$REFG= 000047          EFILLT  020655          F$CLEA= 000007          IXE    = 004000 G
CMDLEN   002366          C$REL  = 000077         EFINIT  017707          F$DU   = 000016         I$AU   = 000041
CMDPAK   002372          C$RESE= 000033          EFINPT  021030          F$END  = 000041         I$AUTO= 000041
CMDRNG   002452          C$REVI= 000003          EFLBFT  020272          F$HARD= 000004          I$CLN  = 000041
CONT     023402          C$RFLA= 000021          EFMEDT  021051          F$HW   = 000013         I$DU   = 000041
CONTON   025640          C$RPT  = 000025         EFNUT   017752          F$INIT= 000006          I$HRD  = 000041
C$AU   = 000052          C$SEFG= 000046          EFRCCT  020203          F$JMP  = 000050         I$INIT= 000041
C$AUTO= 000061           C$SPRI= 000041          EFRCFT  020505          F$MOD  = 000000         I$MOD  = 000041
C$BRK  = 000022          C$SVEC= 000037          EFRCRT  020436          F$MSG  = 000011         I$MSG  = 000041
C$BSEG= 000004           C$TOME= 000076          EFRCVT  017632          F$PROT= 000021          I$PROT= 000040
C$BSUB= 000002           DATARE   002472         EFRCWT  020461          F$PWR  = 000017         I$PTAB= 000041
C$CLCK= 000062           DEFQUE= 000002          EFSEKT  020164          F$RPT  = 000012         I$PWR  = 000041
C$CLEA= 000012           DFBAD    016500         EFSNDT  017753          F$SEG  = 000003         I$RPT  = 000041
C$CLOS= 000035           DFCON    016600         EFSTAT  017524          F$SOFT= 000005          I$SEG  = 000041
C$CLP1= 000006           DFDWN    016550         EFTMOT  020626          F$SRV  = 000010         I$SETU= 000040
C$CPBF= 000074           DFPTBL   002240 G       EFWART  020727          F$SUB  = 000002         I$SRV  = 000041
C$CPME= 000075           DFQSTN   027702         EF.CON= 000036 G        F$SW   = 000014         I$SUB  = 000041
C$CVEC= 000036           DFUNT    016437         EF.NEW= 000035 G        F$TEST= 000001          I$TST  = 000041
C$DCLN= 000044           DF1      007555         EF.PWR= 000034 G        GDSO    021164          J$JMP  = 000167
```

Symbol table

```
LOCAL   002250          L10000  002246          PB11S0  013576          QFSER   016667          TBQ10   006240
LOE   = 040000 G        L10002  025760          PB11S1  013623          QFUIT   016367          TBQ11   006263
LOGUNI  002246          L10003  025770          PB11S2  013655          QNBRA   027652          TBQ12   006312
LOPRGI  002466          L10004  026000          PB11S3  013713          QNBRO   027562          TBQ13   006351
LOT   = 000010 G        L10005  026026          PB11S4  013750          QNBR7   027620          TBQ14   006363
LSTCMD  002462          L10006  032406          PB11S5  014004          QSTN    027544          TBQ15   006402
LSTCRN  002460          L10007  032432          PB11S6  014033          QUESTI= 000001          TBQ16   006413
LSTVCT  002464          L10010  032442          PB11S9  014060          RCD5    027310          TBQ17   006440
L$ACP   002110 G        L10012  032450          PB12    016133          RFDJ4   027204          TBQ18   006457
L$APT   002036 G        MAXDRV= 000004          PB1201  014123          RFD0    021334          TBQ19   006476
L$AUT   002070 G        MOD1    002000 G        PB1202  014207          RFD2    026666          TBQ2    006020
L$AUTO  025762 G        MRQDX1= 000007          PB1203  014274          RFD3    027046          TBQ20   006531
L$CCP   002106 G        MRQDX3= 000023          PB1204  014345          RFD4    027252          TBQ21   006561
L$CLEA  025772 G        MSGNBR= 170000          PB1205  014406          RFD5    027502          TBQ22   006613
L$CO    002032 G        NEXT    025574          PB1206  014447          RFD6    030316          TBQ23   006626
L$DEPO  002011 G        OP.ABR= 000006          PB1207  014521          RFD7    032344          TBQ24   006641
L$DESC  002126 G        OP.ELP= 000003          PB1208  014574          RINTR   002444          TBQ25   006654
L$DESP  002076 G        OP.END= 000200          PB1209  014630          RSPCHK  023516          TBQ26   006667
L$DEVP  002060 G        OP.ESP= 000002          PB1210  014731          RSPPAK  002272          TBQ28   006701
L$DISP  002124 G        OP.GDS= 000001          PB1211  014773          RSPRNG  002446          TBQ29   006731
L$DLY   002116 G        OP.REC= 000005          PB1212  015027          RSP1    002266          TBQ3    006042
L$DTP   002040 G        OP.SEN= 000004          PB1213  015104          SDT6    030124          TBQ30   006762
L$DTYP  002034 G        O$APTS= 000000          PB1214  015150          SDT7    032152          TBQ31   007010
L$DU    026002 G        O$AU  = 000000          PB1215  015221          SETUP   025566          TBQ32   007052
L$DUT   002072 G        O$BGNR= 000000          PB1216  015262          SFBEGT  016740          TBQ33   007105
L$DVTY  002160 G        O$BGNS= 000000          PB1217  015356          SFCYLT  017424          TBQ34   007152
L$EF    002052 G        O$DU  = 000001          PB1218  015453          SFDBBT  017206          TBQ35   007217
L$ENVI  002044 G        O$ERRT= 000000          PB1219  015530          SFDONT  016761          TBQ36   007264
L$ETP   002102 G        O$GNSW= 000000          PB1220  015567          SFFCNT  017477          TBQ37   007331
L$EXP1  002046 G        O$POIN= 000001          PB1221  015654          SFFCUT  017445          TBQ38   007376
L$EXP4  002064 G        O$SETU= 000000          PB1222  015723          SFRBBT  017411          TBQ39   007443
L$EXP5  002066 G        PBF0    012036          PB1223  016016          SFRCBT  017126          TBQ4    006064
L$HARD  032412 G        PBF1    012136          PB13    011746          SFREVT  017005          TBQ40   007510
L$HIME  002120 G        PBF10   013071          PB3     011132          SFR1T   017027          TBQ5    006106
L$HPCP  002016 G        PBF2    012265          PB4     011200          SFR2T   017061          TBQ6    006130
L$HPTP  002022 G        PBF3    012341          PB5     011252          SFTRYT  017346          TBQ7    006152
L$HW    002240 G        PBF4    012435          PB6     011343          SFT0    010707          TBQ8    006174
L$ICP   002104 G        PBF5    012500          PB7     011445          SFT1    010760          TBQ9    006216
L$INIT  025536 G        PBF6    012545          PB8     011477          SFXBBT  017266          TERM    030446
L$LADP  002026 G        PBF7    012642          PB9     011533          SF0     010501          TERMIN= 000004
L$LAST  032436 G        PBF8    012741          PF2     011751          SF1     010550          TIMOUT  026434
L$LOAD  002100 G        PBF9    013031          PLOC    002252          SF100   010611          TNBR4   030550
L$LUN   002074 G        PBSF0   016162          PNT   = 001000 G        SPCL    032052          TNBR12  030464
L$MREV  002050 G        PB0     011021          POLLWT  021074          SPECL = 000006          TNBR13  030516
L$NAME  002000 G        PB1     011050          PRGNAM  002616          SP2INT  026142          TYPASC  016270
L$PRIO  002042 G        PB10    011575          PRI   = 002000 G        SP3INT  026216          TYPDUP  023404
L$PROT  025530 G        PB11    011637          PRI0  = 000000 G        SP4INT  026262          TYPE  = 177760
L$PRT   002112 G        PB11AP  013554          PRI01 = 000040 G        STDALN= 000001          T$ARGC= 000001
L$REPP  002062 G        PB11CR  013131          PRI02 = 000100 G        SVCGBL= 000000          T$CODE= 001031
L$REV   002010 G        PB11EL  013453          PRI03 = 000140 G        SVCINS= 177777          T$ERRN= 000000
L$SPC   002056 G        PB11EN  013307          PRI04 = 000200 G        SVCSUB= 177777          T$EXCP= 000000
L$SPCP  002020 G        PB11ES  013416          PRI05 = 000240 G        SVCTAG= 177777          T$GMAN= 000000
L$SPTP  002024 G        PB11GD  013366          PRI06 = 000300 G        SVCTST= 177777          T$HILI= 000776
L$STA   002030 G        PB11OP  013201          PRI07 = 000340 G        S$LSYM= 010000          T$LAST= 000001
L$TEST  002114 G        PB11RD  013527          PRNTPK  023572          TBLBLD  022174          T$LOLI= 000000
L$TIML  002014 G        PB11SD  013505          PTBL    002254          TBQ0    005711          T$LSYM= 010000
L$UNIT  002012 G        PB11ST  013253          QFDAT   016406          TBQ1    005776          T$LTNO= 000001
```

Symbol table

```
T$NEST= 177777          T$TEMP= 000000          T$$PC  = 000001          UIT0    003000          VECTOR  002262
T$NS0  = 000000         T$TEST= 000001          T$$PRO= 010001          UIT1    003144          VEC.AD  004573
T$NS1  = 000004         T$TSTM= 177777          T$$PTA= 010011          UIT2    003310          WRNGST  026474
T$PCNT= 000000          T$TSTS= 000001          T$$TES= 010006          UIT3    003454          X$ALWA= 000000
T$PTAB= 010011          T$$AUT= 010003          T1      026030 G        UIT4    003620          X$FALS= 000040
T$PTNU= 000001          T$$CLE= 010004          UAM    = 000200 G       UIT5    003764          X$OFFS= 000400
T$SAVL= 177777          T$$DAT= 010012          UIN     002264          UIT6    004130          X$TRUE= 000020
T$SEGL= 177777          T$$DU  = 010005         UITADR  002256          UIT7    004274          $2      025714
T$SUBN= 000000          T$$HAR= 010007          UITDF   004440          UNKWN   032362          $3      025734
T$TAGL= 177777          T$$HW  = 010000         UITOTH= 000010          UNT.NB  005601          $4      025750
T$TAGN= 010013          T$$INI= 010002          UITSIZ= 000144

. ABS.  032450    000   (RW,I,GBL,ABS,OVR)
        000000    001   (RW,I,LCL,REL,CON)
Errors detected:  0

*** Assembler statistics


Work  file  reads: 322
Work  file writes: 319
Size of work file: 38400 Words  ( 150 Pages)
Size of core pool: 19402 Words  ( 74 Pages)
Operating  system: RSX-11M/PLUS (Under VAX/VMS)

Elapsed time: 00:07:09.53
ZRQCA0,ZRQCA0.LST/CR/-SP=SVC35R.MLB/ML,ZRQCA0.MAC
```