```
EEEEEEEEEEEEEEE  RRRRRRRRRRRR     FFFFFFFFFFFFFFF
EEEEEEEEEEEEEEE  RRRRRRRRRRRR     FFFFFFFFFFFFFFF
EEEEEEEEEEEEEEE  RRRRRRRRRRRR     FFFFFFFFFFFFFFF
EEE              RRR         RRR  FFF
EEE              RRR         RRR  FFF
EEE              RRR         RRR  FFF
EEE              RRR         RRR  FFF
EEE              RRR         RRR  FFF
EEE              RRR         RRR  FFF
EEEEEEEEEEEE     RRRRRRRRRRRR     FFFFFFFFFFFF
EEEEEEEEEEEE     RRRRRRRRRRRR     FFFFFFFFFFFF
EEEEEEEEEEEE     RRRRRRRRRRRR     FFFFFFFFFFFF
EEE              RRR   RRR        FFF
EEE              RRR   RRR        FFF
EEE              RRR   RRR        FFF
EEE              RRR      RRR     FFF
EEE              RRR      RRR     FFF
EEE              RRR      RRR     FFF
EEEEEEEEEEEEEEE  RRR         RRR  FFF
EEEEEEEEEEEEEEE  RRR         RRR  FFF
EEEEEEEEEEEEEEE  RRR         RRR  FFF
```

```
_$
Va
--
00
00
00
00
00
00
00
00
00
00
00
00
00
00
00
00
00
00
00
00
00
00
00
000
00
000
000
000
000
000
000
000
000
000
000
000
000
00
00
000
00
00
00
00
00
00
00
00
00
00
00
00
00
```

FILEID**DTAILS

```
DDDDDDDD    TTTTTTTTTT    AAAAAA      IIIIII    LL            SSSSSSSS
DDDDDDDD    TTTTTTTTTT    AAAAAA      IIIIII    LL            SSSSSSSS
DD      DD      TT      AA      AA      II      LL          SS
DD      DD      TT      AA      AA      II      LL          SS
DD      DD      TT      AA      AA      II      LL          SS
DD      DD      TT      AA      AA      II      LL          SS
DD      DD      TT      AA      AA      II      LL            SSSSSS
DD      DD      TT      AA      AA      II      LL            SSSSSS
DD      DD      TT      AAAAAAAAAA      II      LL                  SS
DD      DD      TT      AAAAAAAAAA      II      LL                  SS
DD      DD      TT      AA      AA      II      LL                  SS      ....
DD      DD      TT      AA      AA      II      LL                  SS      ....
DDDDDDDD        TT      AA      AA    IIIIII    LLLLLLLLLL  SSSSSSSS        ....
DDDDDDDD        TT      AA      AA    IIIIII    LLLLLLLLLL  SSSSSSSS        ....

LL              IIIIII      SSSSSSSS
LL              IIIIII      SSSSSSSS
LL                II      SS
LL                II      SS
LL                II      SS
LL                II      SS
LL                II        SSSSSS
LL                II        SSSSSS
LL                II              SS
LL                II              SS
LL                II              SS
LL                II              SS
LLLLLLLLLL      IIIIII    SSSSSSSS
LLLLLLLLLL      IIIIII    SSSSSSSS
```

```
0001    C
0002    C Version:      'V04-000'
0003    C
0004    C*****************************************************************************
0005    C*                                                                           *
0006    C*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                  *
0007    C*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                   *
0008    C*  ALL RIGHTS RESERVED.                                                     *
0009    C*                                                                           *
0010    C*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED    *
0011    C*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE   *
0012    C*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER    *
0013    C*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY    *
0014    C*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY    *
0015    C*  TRANSFERRED.                                                             *
0016    C*                                                                           *
0017    C*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE    *
0018    C*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT    *
0019    C*  CORPORATION.                                                             *
0020    C*                                                                           *
0021    C*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS    *
0022    C*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                  *
0023    C*                                                                           *
0024    C*                                                                           *
0025    C*****************************************************************************
0026    C
0027
0028    c       Author  Brian Porter                    Creation Date   03-NOV-1981
0029
0030    c++
0031    c       Functional description:
0032    c
0033    c       This a collection of routines that are called to output
0034    c       software information in device error entries.
0035    c
0036    c       Modified by:
0037    c
0038    C       V03-003 SAR0214         Sharon A. Reynolds,             28-Mar-1984
0039    C               Changed the name/output label of UCB$L_OWNUIC to
0040    C               ORB$L_OWNER.
0041    C
0042    c       v03-002 BP0003          Brian Porter,                   20-AUG-1982
0043    c               Minor edit.
0044    c
0045    c       v03-001 BP0002          Brian Porter,                   01-MAR-1982
0046    c               Added more routines.
0047    c--
0048    c
0049
0050
0051
0052
0053            subroutine ucb$b_ertcnt (lun,emb$b_dv_ertcnt)
0054
0055
0056
0057
```

```
0058            byte            lun
0059
0060            byte            emb$b_dv_ertcnt
0061
0062            integer*4       remaining_retries
0063
0064            integer*4       compress4
0065
0066
0067
0068            remaining_retries = lib$extzv(0,8,emb$b_dv_ertcnt)
0069
0070            call linchk (lun,2)
0071
0072            write(lun,10) emb$b_dv_ertcnt,remaining_retries
0073    10      format(' ',t8,'UCB$B_ERTCNT',t30,z2.2,/
0074           1 t40,i<compress4 (remaining_retries)>,'. RETRIES REMAINING')
0075
0076            return
0077
0078            end
```

PROGRAM SECTIONS

	Name	Bytes	Attributes
0	$CODE	96	PIC CON REL LCL SHR EXE RD NOWRT LONG
1	$PDATA	67	PIC CON REL LCL SHR NOEXE RD NOWRT LONG
2	$LOCAL	40	PIC CON REL LCL NOSHR NOEXE RD WRT LONG
	Total Space Allocated	203	

ENTRY POINTS

Address	Type	Name
0-00000000		UCB$B_ERTCNT

VARIABLES

Address	Type	Name	Address	Type	Name
AP-00000008@	L*1	EMB$B_DV_ERTCNT	AP-00000004@	L*1	LUN
2-00000000	I*4	REMAINING_RETRIES			

LABELS

Address	Label
1-0000000C	10'

FUNCTIONS AND SUBROUTINES REFERENCED

Type	Name	Type	Name	Type	Name
I*4	COMPRESS4	I*4	LIB$EXTZV		LINCHK

```
0001
0002
0003
0004
0005            subroutine ucb$b_ertmax (lun,emb$b_dv_ertmax)
0006
0007
0008
0009
0010            byte            lun
0011
0012            byte            emb$b_dv_ertmax
0013
0014            integer*4       maximum_retries
0015
0016            integer*4       compress4
0017
0018
0019
0020            maximum_retries = lib$extzv(0,8,emb$b_dv_ertmax)
0021
0022            call linchk (lun,2)
0023
0024            write(lun,10) emb$b_dv_ertmax,maximum_retries
0025    10      format(' ',t8,'UCB$B_ERTMAX',t30,z2.2,/,
0026           1 t40,i<compress4 (maximum_retries)>,' RETRIES ALLOWABLE')
0027
0028            return
0029
0030            end
```

PROGRAM SECTIONS

	Name	Bytes	Attributes
0	$CODE	96	PIC CON REL LCL SHR EXE RD NOWRT LONG
1	$PDATA	67	PIC CON REL LCL SHR NOEXE RD NOWRT LONG
2	$LOCAL	40	PIC CON REL LCL NOSHR NOEXE RD WRT LONG
	Total Space Allocated	203	

ENTRY POINTS

Address	Type	Name
0-00000000		UCB$B_ERTMAX

VARIABLES

Address	Type	Name	Address	Type	Name
AP-00000008@	L*1	EMB$B_DV_ERTMAX	AP-00000004@	L*1	LUN
2-00000000	I*4	MAXIMUM_RETRIES			

LABELS

Address	Label
1-0000000C	10'

FUNCTIONS AND SUBROUTINES REFERENCED

Type	Name	Type	Name	Type	Name
I*4	COMPRESS4	I*4	LIB$EXTZV		LINCHK

```
0001
0002
0003
0004
0005            subroutine ORB$L_OWNER (lun,emb$l_dv_ownuic)
0006
0007
0008
0009
0010            byte            lun
0011
0012            integer*4       emb$l_dv_ownuic
0013
0014            integer*4       group
0015
0016            integer*4       member
0017
0018
0019
0020
0021            group = lib$extzv (16,8,emb$l_dv_ownuic)
0022
0023            member = lib$extzv (0,8,emb$l_dv_ownuic)
0024
0025            call linchk (lun,2)
0026
0027            write(lun,10) emb$l_dv_ownuic,group,member
0028    10      format(' ',t8,'ORB$L_OWNER',t24,z8.8,/,
0029           1 t40,'OWNER UIC [',o3.3,',',o3.3,']')
0030
0031            return
0032
0033            end
```

PROGRAM SECTIONS

	Name	Bytes	Attributes
0	$CODE	109	PIC CON REL LCL SHR EXE RD NOWRT LONG
1	$PDATA	68	PIC CON REL LCL SHR NOEXE RD NOWRT LONG
2	$LOCAL	52	PIC CON REL LCL NOSHR NOEXE RD WRT LONG
	Total Space Allocated	229	

ENTRY POINTS

Address	Type	Name
0-00000000		ORB$L_OWNER

VARIABLES

Address	Type	Name	Address	Type	Name
AP-00000008@	I*4	EMB$L_DV_OWNUIC	2-00000000	I*4	GROUP
AP-00000004@	L*1	LUN	2-00000004	I*4	MEMBER

LABELS

Address	Label
1-00000010	10'

FUNCTIONS AND SUBROUTINES REFERENCED

Type	Name	Type	Name
I*4	LIB$EXTZV		LINCHK

```
0001
0002
0003
0004
0005            subroutine ucb$l_opcnt (lun,emb$l_dv_opcnt)
0006
0007
0008
0009
0010            byte            lun
0011
0012            integer*4       emb$l_dv_opcnt
0013
0014            integer*4       compress4
0015
0016
0017
0018
0019            call linchk (lun,2)
0020
0021            write(lun,10) (emb$l_dv_opcnt,i = 1,2)
0022    10      format(' ',t8,'UCB$L_OPCNT',t24,z8.8,/,
0023            1 t40,i<compress4 (emb$l_dv_opcnt)>,17h. QIO'S THIS UNIT)
0024
0025            return
0026
0027            end
```

PROGRAM SECTIONS

	Name	Bytes	Attributes
0	$CODE	97	PIC CON REL LCL SHR EXE RD NOWRT LONG
1	$PDATA	56	PIC CON REL LCL SHR NOEXE RD NOWRT LONG
2	$LOCAL	28	PIC CON REL LCL NOSHR NOEXE RD WRT LONG
	Total Space Allocated	181	

ENTRY POINTS

Address	Type	Name
0-00000000		UCB$L_OPCNT

VARIABLES

Address	Type	Name	Address	Type	Name
2-00000004@	I*4	EMB$L_DV_OPCNT	2-00000000	I*4	I
AP-00000004@	L*1	LUN			

LABELS

Address	Label
1-00000004	10'

FUNCTIONS AND SUBROUTINES REFERENCED

Type	Name	Type	Name
I*4	COMPRESS4		LINCHK

```
0001
0002
0003
0004
0005            subroutine ucb$w_errcnt (lun,emb$w_dv_errcnt)
0006
0007
0008
0009
0010            byte            lun
0011
0012            integer*2       emb$w_dv_errcnt
0013
0014            integer*4       errors_this_unit
0015
0016            integer*4       compress4
0017
0018
0019
0020
0021            errors_this_unit = lib$extzv (0,16,emb$w_dv_errcnt)
0022
0023            call linchk (lun,2)
0024
0025            write(lun,10) emb$w_dv_errcnt,errors_this_unit
0026    10      format(' ',t8,'UCB$W_ERRCNT',t28,z4.4,/,
0027           1 t40,i<compress4 (errors_this_unit)>,'. ERRORS THIS UNIT')
0028
0029            return
0030
0031            end
```

PROGRAM SECTIONS

	Name	Bytes	Attributes
0	$CODE	96	PIC CON REL LCL SHR EXE RD NOWRT LONG
1	$PDATA	66	PIC CON REL LCL SHR NOEXE RD NOWRT LONG
2	$LOCAL	40	PIC CON REL LCL NOSHR NOEXE RD WRT LONG
	Total Space Allocated	202	

ENTRY POINTS

Address	Type	Name
0-00000000		UCB$W_ERRCNT

VARIABLES

Address	Type	Name	Address	Type	Name
AP-00000008@	I*2	EMB$W_DV_ERRCNT	2-00000000	I*4	ERRORS_THIS_UNIT
AP-00000004@	L*1	LUN			

LABELS

Address	Label
1-0000000C	10'

FUNCTIONS AND SUBROUTINES REFERENCED

Type	Name	Type	Name	Type	Name
I*4	COMPRESS4	I*4	LIB$EXTZV		LINCHK

```
0001
0002
0003
0004
0005          subroutine ucb$l_char (lun,emb$l_dv_char)
0006
0007
0008
0009
0010          byte            lun
0011
0012          integer*4       emb$l_dv_char
0013
0014          character*33    v1device_characteristics(0:7)
0015
0016          data    v1device_characteristics(0)  /'RECORD ORIENTED*'/
0017
0018          data    v1device_characteristics(1)  /'CARRIAGE CONTROL*'/
0019
0020          data    v1device_characteristics(2)  /'TERMINAL*'/
0021
0022          data    v1device_characteristics(3)  /'DIRECTORY STRUCTURED*'/
0023
0024          data    v1device_characteristics(4)  /'SINGLE DIRECTORY*'/
0025
0026          data    v1device_characteristics(5)  /'"SEQUENTIAL BLOCK" ORIENTED*'/
0027
0028          data    v1device_characteristics(6)  /'SPOOLED*'/
0029
0030          data    v1device_characteristics(7)  /'OPERATOR*'/
0031
0032          character*22    v2device_characteristics(13:31)
0033
0034          data    v2device_characteristics(13) /'NETWORK*'/
0035
0036          data    v2device_characteristics(14) /'FILE ORIENTED*'/
0037
0038          data    v2device_characteristics(15) /'DUAL ACCESS*'/
0039
0040          data    v2device_characteristics(16) /'SHARABLE*'/
0041
0042          data    v2device_characteristics(17) /'GEN*'/
0043
0044          data    v2device_characteristics(18) /'AVAILABLE*'/
0045
0046          data    v2device_characteristics(19) /'MOUNTED*'/
0047
0048          data    v2device_characteristics(20) /'MAILBOX*'/
0049
0050          data    v2device_characteristics(21) /'"MARKED" FOR DISMOUNT*'/
0051
0052          data    v2device_characteristics(22) /'ERROR LOGGING*'/
0053
0054          data    v2device_characteristics(23) /'ALLOCATED*'/
0055
0056          data    v2device_characteristics(24) /'FOREIGN*'/
0057
```

```
0058            data    v2device_characteristics(25) /'SOFTWARE "WRITE LOCK"*'/
0059
0060            data    v2device_characteristics(26) /'CAPABLE OF INPUT*'/
0061
0062            data    v2device_characteristics(27) /'CAPABLE OF OUTPUT*'/
0063
0064            data    v2device_characteristics(28) /'RANDOM ACCESS*'/
0065
0066            data    v2device_characteristics(29) /'REAL TIME*'/
0067
0068            data    v2device_characteristics(30) /'DATA CHECK, READS*'/
0069
0070            data    v2device_characteristics(31) /'DATA CHECK, WRITES*'/
0071
0072
0073
0074            call linchk (lun,1)
0075
0076            write(lun,10) emb$l_dv_char
0077    10      format(' ',t8,'UCB$L_CHAR',t24,z8.8)
0078
0079            call output (lun,emb$l_dv_char,v1device_characteristics,0,0,6,'0')
0080
0081            call output (lun,emb$l_dv_char,v2device_characteristics,
0082            1 13,13,31,'0')
0083
0084            return
0085
0086            end
```

PROGRAM SECTIONS

	Name	Bytes	Attributes
0	$CODE	93	PIC CON REL LCL SHR EXE RD NOWRT LONG
1	$PDATA	45	PIC CON REL LCL SHR NOEXE RD NOWRT LONG
2	$LOCAL	832	PIC CON REL LCL NOSHR NOEXE RD WRT LONG
	Total Space Allocated	970	

ENTRY POINTS

Address	Type	Name
0-00000000		UCB$L_CHAR

VARIABLES

Address	Type	Name	Address	Type	Name
AP-00000008@	I*4	EMB$L_DV_CHAR	AP-00000004@	L*1	LUN

ARRAYS

Address	Type	Name	Bytes	Dimensions
2-00000000	CHAR	V1DEVICE_CHARACTERISTICS	264	(0:7)
2-00000108	CHAR	V2DEVICE_CHARACTERISTICS	418	(13:31)

LABELS

Address	Label
1-00000016	10'

FUNCTIONS AND SUBROUTINES REFERENCED

Type	Name	Type	Name
	LINCHK		OUTPUT

```
0001
0002
0003
0004
0005            subroutine ucb$w_sts (lun,emb$w_dv_sts)
0006
0007
0008
0009
0010            byte            lun
0011
0012            integer*2       emb$w_dv_sts
0013
0014            character*26    v1device_status(0:12)
0015
0016            data    v1device_status(0)      /'TIMEOUT ENABLED*'/
0017
0018            data    v1device_status(1)      /'INTERRUPT EXPECTED*'/
0019
0020            data    v1device_status(2)      /'ERROR LOGGING IN-PROGRESS*'/
0021
0022            data    v1device_status(3)      /'CANCEL I/O*'/
0023
0024            data    v1device_status(4)      /'ONLINE*'/
0025
0026            data    v1device_status(5)      /'POWER FAILED WHILE BUSY*'/
0027
0028            data    v1device_status(6)      /'TIMED OUT*'/
0029
0030            data    v1device_status(7)      /'RECEIVER INTERRUPT*'/
0031
0032            data    v1device_status(8)      /'BUSY*'/
0033
0034            data    v1device_status(9)      /'MOUNT IN-PROGRESS*'/
0035
0036            data    v1device_status(10)     /'DEALLOCATE AT DISMOUNT*'/
0037
0038            data    v1device_status(11)     /'SOFTWARE VALID*'/
0039
0040            data    v1device_status(12)     /'UNLOAD AT DISMOUNT*'/
0041
0042            character*33    v2device_status(14:15)
0043
0044            data    v2device_status(14)     /''MOUNT VERIFICATION'' IN-PROGRESS*'/
0045
0046            data    v2device_status(15)     /'WRONG VOLUME DETECTED*'/
0047
0048
0049
0050            call linchk (lun,1)
0051
0052            write(lun,10) emb$w_dv_sts
0053    10      format(' ',t8,'UCB$W_STS',t28,z4.4)
0054
0055            call output (lun,emb$w_dv_sts,v1device_status,0,0,12,'0')
0056
0057            call output (lun,emb$w_dv_sts,v2device_status,14,14,15,'0')
```

```
0058
0059            return
0060
0061            end
```

PROGRAM SECTIONS

	Name	Bytes	Attributes
0	$CODE	93	PIC CON REL LCL SHR EXE RD NOWRT LONG
1	$PDATA	44	PIC CON REL LCL SHR NOEXE RD NOWRT LONG
2	$LOCAL	552	PIC CON REL LCL NOSHR NOEXE RD WRT LONG
	Total Space Allocated	689	

ENTRY POINTS

Address	Type	Name
0-00000000		UCB$W_STS

VARIABLES

Address	Type	Name	Address	Type	Name
AP-00000008@	I*2	EMB$W_DV_STS	AP-00000004@	L*1	LUN

ARRAYS

Address	Type	Name	Bytes	Dimensions
2-00000000	CHAR	V1DEVICE_STATUS	338	(0:12)
2-00000152	CHAR	V2DEVICE_STATUS	66	(14:15)

LABELS

Address	Label
1-00000016	10'

FUNCTIONS AND SUBROUTINES REFERENCED

Type	Name	Type	Name
	LINCHK		OUTPUT

```
0001
0002
0003
0004
0005            subroutine ucb$l_media (lun,emb$l_dv_media)
0006
0007
0008
0009
0010            byte            lun
0011
0012            integer*4       emb$l_dv_media
0013
0014            integer*4       cylinder
0015
0016            integer*4       track
0017
0018            integer*4       sector
0019
0020            integer*4       compress4
0021
0022
0023
0024
0025            cylinder = lib$extzv (16,16,emb$l_dv_media)
0026
0027            track = lib$extzv (8,8,emb$l_dv_media)
0028
0029            sector = lib$extzv (0,8,emb$l_dv_media)
0030
0031            call linchk (lun,5)
0032
0033            write(lun,10) emb$l_dv_media,cylinder,track,sector
0034    10      format(' ',t8,'UCB$[_MEDIA',t24,z8.8,/,
0035           1 t40,'FUNCTION START ADDRESS.',/,
0036           1 t40,' - CYLINDER #',i<compress4 (cylinder)>,'..',/,
0037           1 t40,' - TRACK #',i<compress4 (track)>,'..',/,
0038           1 t40,' - SECTOR #',i<compress4 (sector)>,'.')
0039
0040            return
0041
0042            end
```

PROGRAM SECTIONS

	Name	Bytes	Attributes
0	$CODE	178	PIC CON REL LCL SHR EXE RD NOWRT LONG
1	$PDATA	146	PIC CON REL LCL SHR NOEXE RD NOWRT LONG
2	$LOCAL	96	PIC CON REL LCL NOSHR NOEXE RD WRT LONG
	Total Space Allocated	420	

ENTRY POINTS

Address	Type	Name
0-00000000		UCB$L_MEDIA

VARIABLES

Address	Type	Name	Address	Type	Name
2-00000000	I*4	CYLINDER	AP-00000008@	I*4	EMB$L_DV_MEDIA
AP-00000004@	L*1	LUN	2-00000008	I*4	SECTOR
2-00000004	I*4	TRACK			

LABELS

Address	Label
1-00000010	10'

FUNCTIONS AND SUBROUTINES REFERENCED

Type	Name	Type	Name	Type	Name
I*4	COMPRESS4	I*4	LIB$EXTZV		LINCHK

```
0001
0002
0003
0004
0005          subroutine irp$w_func (lun,emb$w_dv_func,qio_string)
0006
0007
0008
0009
0010          byte            lun
0011
0012          integer*2       emb$w_dv_func
0013
0014          character*(*)   qio_string
0015
0016          integer*4       device_dependent_modifiers
0017
0018          character*15    v1device_independent_modifiers(14:15)
0019
0020          data            v1device_independent_modifiers(14)
0021         1 /'IO$M_DATACHECK*'/
0022
0023          data            v1device_independent_modifiers(15)
0024         1 /'IO$M_INHRETRY*'/
0025
0026          integer*4       compressc
0027
0028
0029
0030
0031          call linchk (lun,2)
0032
0033          write(lun,10) emb$w_dv_func,qio_string
0034     10   format(' ',t8,'IRP$W_FUNC',t28,z4.4,/,
0035         1 t40,a<compressc (qio_string)>)
0036
0037
0038
0039          entry qio_function_modifiers (lun,emb$w_dv_func)
0040
0041
0042
0043          device_dependent_modifiers = lib$extzv(6,7,emb$w_dv_func)
0044
0045          if (device_dependent_modifiers .ne. 0) then
0046
0047          call linchk (lun,1)
0048
0049          write(lun,15)
0050     15   format(' ',t40,'DEVICE DEPENDENT MODIFIER(S)')
0051          endif
0052
0053          call output (lun,emb$w_dv_func,v1device_independent_modifiers,
0054          1 14,14,15,'0')
0055
0056          return
0057
```

```
0058          end


PROGRAM SECTIONS

    Name                            Bytes   Attributes

  0 $CODE                             182   PIC CON REL LCL   SHR   EXE   RD NOWRT LONG
  1 $PDATA                             94   PIC CON REL LCL   SHR NOEXE   RD NOWRT LONG
  2 $LOCAL                            168   PIC CON REL LCL NOSHR NOEXE   RD   WRT LONG

    Total Space Allocated             444

ENTRY POINTS

    Address  Type  Name                        Address  Type  Name

  0-00000000       IRP$W_FUNC                0-00000055       QIO_FUNCTION_MODIFIERS

VARIABLES

    Address  Type  Name                        Address  Type  Name

  2-00000020  I*4  DEVICE_DEPENDENT_MODIFIERS  AP-00000008@ I*2  EMB$W_DV_FUNC
 AP-00000004@ L*1  LUN                         2-00000024@ CHAR QIO_STRING

ARRAYS

    Address  Type  Name                                 Bytes  Dimensions

  2-00000000  CHAR V1DEVICE_INDEPENDENT_MODIFIERS          30  (14:15)

LABELS

    Address   Label       Address   Label

  1-0000001A  10'       1-0000003A  15'

FUNCTIONS AND SUBROUTINES REFERENCED

  Type  Name            Type  Name            Type  Name            Type  Name

   I*4  COMPRESSC        I*4  LIB$EXTZV             LINCHK                OUTPUT
```

```
0001
0002
0003
0004
0005            subroutine irp$w_bcnt (lun,emb$w_dv_bcnt)
0006
0007
0008
0009
0010            byte            lun
0011
0012            integer*2       emb$w_dv_bcnt
0013
0014            integer*4       transfer_byte_count
0015
0016            integer*4       compress4
0017
0018
0019
0020
0021            transfer_byte_count = lib$extzv(0,16,emb$w_dv_bcnt)
0022
0023            call linchk (lun,2)
0024
0025            write(lun,10) emb$w_dv_bcnt,transfer_byte_count
0026    10      format(' ',t8,'IRP$W_BCNT',t28,z4.4,7,
0027           1 t40,'TRANSFER SIZE ',i<compress4 (transfer_byte_count)>,'. BYTE(S)')
0028
0029            return
0030
0031            end
```

CDR

PRO

0
1
2

ENT

0

VAR

AP

LAB

1·

FUN

T·

PROGRAM SECTIONS

	Name	Bytes	Attributes
0	$CODE	96	PIC CON REL LCL SHR EXE RD NOWRT LONG
1	$PDATA	71	PIC CON REL LCL SHR NOEXE RD NOWRT LONG
2	$LOCAL	40	PIC CON REL LCL NOSHR NOEXE RD WRT LONG
	Total Space Allocated	207	

ENTRY POINTS

Address	Type	Name
0-00000000		IRP$W_BCNT

VARIABLES

Address	Type	Name	Address	Type	Name
AP-00000008@	I*2	EMB$W_DV_BCNT	AP-00000004@	L*1	LUN
2-00000000	I*4	TRANSFER_BYTE_COUNT			

LABELS

Address	Label
1-0000000C	10'

FUNCTIONS AND SUBROUTINES REFERENCED

Type	Name	Type	Name	Type	Name
I*4	COMPRESS4	I*4	LIB$EXTZV		LINCHK

```
0001
0002
0003
0004            subroutine irp$w_boff (lun,emb$w_dv_boff)
0005
0006
0007
0008            byte            lun
0009
0010            integer*2       emb$w_dv_boff
0011
0012            integer*4       page_byte_offset
0013
0014            integer*4       compress4
0015
0016
0017
0018            page_byte_offset = lib$extzv (0,16,emb$w_dv_boff)
0019
0020            call linchk (lun,2)
0021
0022            if (page_byte_offset .eq. 0) then
0023
0024            write(lun,10) emb$w_dv_boff
0025    10      format(' ',t8,'IRP$W_BOFF',t28,z4.4,/,
0026            1 t40,'TRANSFER PAGE ALIGNED')
0027            else
0028
0029            write(lun,15) emb$w_dv_boff,page_byte_offset
0030    15      format(' ',t8,'IRP$W_BOFF',t28,z4.4,/,
0031            1 t40,i<compress4 (page_byte_offset)>,'. BYTE PAGE OFFSET')
0032            endif
0033
0034            return
0035
0036            end
```

PROGRAM SECTIONS

	Name	Bytes	Attributes
0	$CODE	136	PIC CON REL LCL SHR EXE RD NOWRT LONG
1	$PDATA	113	PIC CON REL LCL SHR NOEXE RD NOWRT LONG
2	$LOCAL	40	PIC CON REL LCL NOSHR NOEXE RD WRT LONG
	Total Space Allocated	289	

ENTRY POINTS

Address	Type	Name
0-00000000		IRP$W_BOFF

VARIABLES

Address	Type	Name	Address	Type	Name
AP-00000008@	I*2	EMB$W_DV_BOFF	AP-00000004@	L*1	LUN
2-00000000	I*4	PAGE_BYTE_OFFSET			

LABELS

Address	Label	Address	Label
1-0000000C	10'	1-0000003D	15'

FUNCTIONS AND SUBROUTINES REFERENCED

Type	Name	Type	Name	Type	Name
I*4	COMPRESS4	I*4	LIBSEXTZV		LINCHK

000
000
000
000
000
000
000
0001
0009
0010
001
001
001
001
0014
001
0016
001
001
001
0019
0020
002
002
002
002
002
002
002
002
002
002
003
003
003

```
0001
0002
0003
0004
0005            subroutine irp$l_pid (lun,emb$l_dv_rqpid)
0006
0007
0008
0009
0010            byte            lun
0011
0012            integer*4       emb$l_dv_rqpid
0013
0014
0015
0016
0017            call linchk (lun,2)
0018
0019            write(lun,10) emb$l_dv_rqpid
0020    10      format(' ',t8,'IRP$L_PID',t24,z8.8,/,
0021            1 t40,'REQUESTOR ''PID''')
0022
0023            return
0024
0025            end
```

PROGRAM SECTIONS

	Name	Bytes	Attributes
0	$CODE	56	PIC CON REL LCL SHR EXE RD NOWRT LONG
1	$PDATA	46	PIC CON REL LCL SHR NOEXE RD NOWRT LONG
2	$LOCAL	12	PIC CON REL LCL NOSHR NOEXE RD WRT LONG
	Total Space Allocated	114	

ENTRY POINTS

Address	Type	Name
0-00000000		IRP$L_PID

VARIABLES

Address	Type	Name	Address	Type	Name
AP-00000008@	I*4	EMB$L_DV_RQPID	AP-00000004@	L*1	LUN

CDR
PRO
0
1
2
ENT
0
VAR
2
AP
ARR
AP
LAB
1
FUN
T

LABELS

Address	Label
1-00000004	10'

FUNCTIONS AND SUBROUTINES REFERENCED

Type	Name
	LINCHK

```
                                                                                000
                                                                                000
                                                                                000
                                                                                000
                                                                                000
                                                                                000
                                                                                000
                                                                                000
                                                                                001
                                                                                001
                                                                                001
                                                                                001
                                                                                001
                                                                                001
                                                                                001
                                                                                001
                                                                                001
                                                                                001
                                                                                002
                                                                                002
                                                                                002
                                                                                002
                                                                                002
                                                                                002
                                                                                002
                                                                                002
                                                                                002
                                                                                002
PRO
                                                                                0
                                                                                1
                                                                                2
ENT
                                                                                0
VAR
                                                                                2
                                                                                AP
```

```
0001
0002
0003
0004
0005            subroutine irp$q_iosb (lun,emb$l_dv_iosb)
0006
0007
0008
0009
0010            byte            lun
0011
0012            integer*4       emb$l_dv_iosb(2)
0013
0014            integer*4       bytes_transfered
0015
0016            integer*4       compress4
0017
0018
0019
0020
0021            bytes_transfered = lib$extzv (16,16,emb$l_dv_iosb(1))
0022
0023            call linchk (lun,2)
0024
0025            write(lun,10) (emb$l_dv_iosb(i),i = 1,2),bytes_transfered
0026    10      format(' ',t8,'IRP$Q_IOSB',t24,z8.8,/,
0027           1 t24,z8.8,t40,'IOSB, ',i<compress4 (bytes_transfered)>,
0028           1 '. BYTE(S) TRANSFERRED')
0029
0030            return
0031
0032            end
```

PROGRAM SECTIONS

	Name	Bytes	Attributes
0	$CODE	122	PIC CON REL LCL SHR EXE RD NOWRT LONG
1	$PDATA	76	PIC CON REL LCL SHR NOEXE RD NOWRT LONG
2	$LOCAL	64	PIC CON REL LCL NOSHR NOEXE RD WRT LONG
	Total Space Allocated	262	

ENTRY POINTS

Address	Type	Name
0-00000000		IRP$Q_IOSB

VARIABLES

Address	Type	Name	Address	Type	Name
2-00000000	I*4	BYTES_TRANSFERED	2-00000004	I*4	I
AP-00000004@	L*1	LUN			

ARRAYS

Address	Type	Name	Bytes	Dimensions
AP-00000008@	I*4	EMB$L_DV_IOSB	8	(2)

LABELS

Address	Label
1-00000008	10'

FUNCTIONS AND SUBROUTINES REFERENCED

Type	Name	Type	Name	Type	Name
I*4	COMPRESS4	I*4	LIB$EXTZV		LINCHK

```
0001
0002
0003
0004
0005            subroutine cdrp$w_boff (lun,emb$w_sp_boff)
0006
0007
0008
0009
0010            byte            lun
0011
0012            integer*2       emb$w_sp_boff
0013
0014            integer*4       page_byte_offset
0015
0016            integer*4       compress4
0017
0018
0019
0020            page_byte_offset = lib$extzv (0,16,emb$w_sp_boff)
0021
0022            call linchk (lun,2)
0023
0024            if (page_byte_offset .eq. 0) then
0025
0026            write(lun,10) emb$w_sp_boff
0027     10     format(' ',t8,'CDRP$W_BOFF',t28,z4.4,/,
0028            1 t40,'TRANSFER PAGE ALIGNED')
0029            else
0030
0031            write(lun,15) emb$w_sp_boff,page_byte_offset
0032     15     format(' ',t8,'CDRP$W_BOFF',t28,z4.4,/,
0033            1 t40,i<compress4 (page_byte_offset)>,'. BYTE PAGE OFFSET')
0034            endif
0035
0036            return
0037
0038            end
```

PROGRAM SECTIONS

	Name	Bytes	Attributes
0	$CODE	136	PIC CON REL LCL SHR EXE RD NOWRT LONG
1	$PDATA	115	PIC CON REL LCL SHR NOEXE RD NOWRT LONG
2	$LOCAL	40	PIC CON REL LCL NOSHR NOEXE RD WRT LONG
	Total Space Allocated	291	

ENTRY POINTS

Address	Type	Name
0-00000000		CDRP$W_BOFF

VARIABLES

Address	Type	Name	Address	Type	Name
AP-00000008@	I*2	EMB$W_SP_BOFF	AP-00000004@	L*1	LUN
2-00000000	I*4	PAGE_BYTE_OFFSET			

LABELS

Address	Label	Address	Label
1-0000000C	10'	1-0000003E	15'

FUNCTIONS AND SUBROUTINES REFERENCED

Type	Name	Type	Name	Type	Name
I*4	COMPRESS4	I*4	LIB$EXTZV		LINCHK

DUM

005
005
006
006
006
006
006
006
006
006
006
007
007
007
007
007
007
007
007
007
007
008
008
008
008
008
008
008
008
008
008
009
009
009
009

```
0001
0002
0003
0004            subroutine cdrp$l_bcnt (lun,emb$l_sp_bcnt)
0005
0006
0007
0008
0009            byte            lun
0010
0011            integer*4       emb$l_sp_bcnt
0012
0013            integer*4       compress4
0014
0015
0016
0017
0018            call linchk (lun,2)
0019
0020            write(lun,10) (emb$l_sp_bcnt,i = 0,1)
0021    10      format(' ',t8,'CDRP$L_BCNT',t24,z8.8,/,
0022            1 t40,'TRANSFER SIZE ',i<compress4 (emb$l_sp_bcnt)>,'. BYTE(S)')
0023
0024            return
0025
0026            end
```

PROGRAM SECTIONS

	Name	Bytes	Attributes
0	$CODE	96	PIC CON REL LCL SHR EXE RD NOWRT LONG
1	$PDATA	64	PIC CON REL LCL SHR NOEXE RD NOWRT LONG
2	$LOCAL	28	PIC CON REL LCL NOSHR NOEXE RD WRT LONG
	Total Space Allocated	188	

ENTRY POINTS

Address	Type	Name
0-00000000		CDRP$L_BCNT

VARIABLES

Address	Type	Name	Address	Type	Name	Address	Type	Name
2-0000004@	I*4	EMB$L_SP_BCNT	2-00000000	I*4	I	AP-0000004@	L*1	LUN

LABELS

Address	Label
1-00000004	10'

FUNCTIONS AND SUBROUTINES REFERENCED

Type	Name	Type	Name
I*4	COMPRESS4		LINCHK

```
0001
0002
0003
0004
0005            subroutine cdrp$w_func (lun,emb$w_sp_func,qio_string)
0006
0007
0008
0009
0010            byte            lun
0011
0012            integer*2       emb$w_sp_func
0013
0014            character*(*)   qio_string
0015
0016            integer*4       compressc
0017
0018
0019
0020
0021            call linchk (lun,2)
0022
0023            write(lun,10) emb$w_sp_func,qio_string
0024    10      format(' ',t8,'CDRP$W_FUNC',t28,z4.4,/,
0025           1 t40,a<compressc (qio_string)>)
0026
0027            call qio_function_modifiers (lun,emb$w_sp_func)
0028
0029            return
0030
0031            end
```

PROGRAM SECTIONS

	Name	Bytes	Attributes
0	$CODE	115	PIC CON REL LCL SHR EXE RD NOWRT LONG
1	$PDATA	37	PIC CON REL LCL SHR NOEXE RD NOWRT LONG
2	$LOCAL	44	PIC CON REL LCL NOSHR NOEXE RD WRT LONG
	Total Space Allocated	196	

ENTRY POINTS

Address	Type	Name
0-00000000		CDRP$W_FUNC

VARIABLES

Address	Type	Name	Address	Type	Name	Address	Type	Name
AP-00000008@	I*2	EMB$W_SP_FUNC	AP-00000004@	L*1	LUN	2-00000000@	CHAR	QIO_STRING

LABELS

Address	Label
1-00000004	10'

FUNCTIONS AND SUBROUTINES REFERENCED

Type	Name	Type	Name	Type	Name
I*4	COMPRESSC		LINCHK		QIO_FUNCTION_MODIFIERS

```
0001
0002
0003
0004            subroutine cdrp$l_pid (lun,emb$l_sp_rqpid)
0005
0006
0007
0008
0009            byte            lun
0010
0011            integer*4       emb$l_sp_rqpid
0012
0013
0014
0015
0016            call linchk (lun,2)
0017
0018            write(lun,10) emb$l_sp_rqpid
0019    10      format(' ',t8,'CDRP$L_PID',t24,z8.8,/,
0020            1 t40,'REQUESTOR "PID"')
0021
0022            return
0023
0024            end
```

PROGRAM SECTIONS

	Name	Bytes	Attributes
0	$CODE	56	PIC CON REL LCL SHR EXE RD NOWRT LONG
1	$PDATA	47	PIC CON REL LCL SHR NOEXE RD NOWRT LONG
2	$LOCAL	12	PIC CON REL LCL NOSHR NOEXE RD WRT LONG
	Total Space Allocated	115	

ENTRY POINTS

Address	Type	Name
0-00000000		CDRP$L_PID

VARIABLES

Address	Type	Name	Address	Type	Name
AP-00000008@	I*4	EMB$L_SP_RQPID	AP-00000004@	L*1	LUN

LABELS

Address	Label
1-00000004	10'

FUNCTIONS AND SUBROUTINES REFERENCED

Type	Name
	LINCHK

DUP

011
011
011
011
011
012
012
012
012
012
012
012
012
012
012
012
013
013
013
013
013
013
013
013
013
013
014
014
014
014
014
014
014
014
014
014
015
015
015
015
015
015
015
015
015
015
015
016
016
016
016
016
016
016
016
016
016
016
016
017
017

```
0001
0002
0003
0004
0005            subroutine cdrp$q_iosb (lun,emb$l_sp_iosb)
0006
0007
0008
0009
0010            byte            lun
0011
0012            integer*4       emb$l_sp_iosb(2)
0013
0014            integer*4       bytes_transfered
0015
0016            integer*4       compress4
0017
0018
0019
0020
0021            bytes_transfered = lib$extzv (16,16,emb$l_sp_iosb(1))
0022
0023            call linchk (lun,2)
0024
0025            write(lun,10) (emb$l_sp_iosb(i),i = 1,2),bytes_transfered
0026    10      format(' ',t8,'CDRP$Q_IOSB',t24,z8.8,/,
0027            1 t24,z8.8,t40,'IOSB, ',i<compress4 (bytes_transfered)>,
0028            1 '. BYTE(S) TRANSFERRED')
0029
0030            return
0031
0032            end
```

PROGRAM SECTIONS

	Name	Bytes	Attributes
0	$CODE	122	PIC CON REL LCL SHR EXE RD NOWRT LONG
1	$PDATA	77	PIC CON REL LCL SHR NOEXE RD NOWRT LONG
2	$LOCAL	64	PIC CON REL LCL NOSHR NOEXE RD WRT LONG
	Total Space Allocated	263	

ENTRY POINTS

Address	Type	Name
0-00000000		CDRP$Q_IOSB

VARIABLES

Address	Type	Name	Address	Type	Name
2-00000000	I*4	BYTES_TRANSFERED	2-00000004	I*4	I
AP-00000004@	L*1	LUN			

ARRAYS

Address	Type	Name	Bytes	Dimensions
AP-00000008@	I*4	EMB$L_SP_IOSB	8	(2)

LABELS

Address	Label
1-00000008	10'

FUNCTIONS AND SUBROUTINES REFERENCED

Type	Name	Type	Name	Type	Name
I*4	COMPRESS4	I*4	LIB$EXTZV		LINCHK

```
0001
0002
0003
0004
0005            subroutine cdrp$l_media (lun,emb$l_sp_media)
0006
0007
0008
0009
0010            byte            lun
0011
0012            integer*4       emb$l_sp_media
0013
0014            integer*4       compress4
0015
0016
0017
0018
0019            call linchk (lun,2)
0020
0021            write(lun,10) (emb$l_sp_media,i = 0,1)
0022    10      format(' ',t8,'CDRP$L_MEDIA',t24,z8.8,/,
0023           1 t40,'STARTING LBN #',i<compress4 (emb$l_sp_media)>,'.')
0024
0025            return
0026
0027            end
```

PROGRAM SECTIONS

	Name	Bytes	Attributes
0	$CODE	96	PIC CON REL LCL SHR EXE RD NOWRT LONG
1	$PDATA	57	PIC CON REL LCL SHR NOEXE RD NOWRT LONG
2	$LOCAL	28	PIC CON REL LCL NOSHR NOEXE RD WRT LONG
	Total Space Allocated	181	

ENTRY POINTS

Address	Type	Name
0-00000000		CDRP$L_MEDIA

VARIABLES

Address	Type	Name	Address	Type	Name
2-00000004@	I*4	EMB$L_SP_MEDIA	2-00000000	I*4	I
AP-00000004@	L*1	LUN			

LABELS

Address	Label
1-00000004	10'

FUNCTIONS AND SUBROUTINES REFERENCED

Type	Name	Type	Name
I*4	COMPRESS4		LINCHK

COMMAND QUALIFIERS

```
FORTRAN /LIS=LIS$:DTAILS/OBJ=OBJ$:DTAILS MSRC$:DTAILS

/CHECK=(NOBOUNDS,OVERFLOW,NOUNDERFLOW)
/DEBUG=(NOSYMBOLS,TRACEBACK)
/STANDARD=(NOSYNTAX,NOSOURCE_FORM)
/SHOW=(NOPREPROCESSOR,NOINCLUDE,MAP)
/F77  /NOG_FLOATING  /I4  /OPTIMIZE  /WARNINGS  /NOD_LINES  /NOCROSS_REFERENCE  /NOMACHINE_CODE  /CONTINUATIONS=19
```

COMPILATION STATISTICS

```
Run Time:            8.21 seconds
Elapsed Time:        20.50 seconds
Page Faults:         147
Dynamic Memory:      170 pages
```

0147 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

CLASSIFY
LIS

DR750
LIS

CSTRING
LIS

DHEADS
LIS

DR780
LIS

DR11W
LIS

COMPRESS
LIS

DTAILS
LIS

DECODECC
LIS

DUMPREG
LIS

DUTUDRIVR
LIS

CALCMAP
LIS

DUP3271
LIS

CRYPTK
LIS

DQDISKS
LIS

DUP11
LIS