K 6

```
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
00
00
00
00
00
00
00
00
00
00
00
00
00
00
00
00
00
00
00
00
00
00
00
00
00
00
00
00
00
```

```
**FILE**ID**ROLLUP                                                  I 13



RRRRRRRR      000000   LL         LL         UU      UU  PPPPPPPP
RRRRRRRR      000000   LL         LL         UU      UU  PPPPPPPP
RR      RR  00      00 LL         LL         UU      UU  PP      PP
RR      RR  00      00 LL         LL         UU      UU  PP      PP
RR      RR  00      00 LL         LL         UU      UU  PP      PP
RR      RR  00      00 LL         LL         UU      UU  PP      PP
RRRRRRRR    00      00 LL         LL         UU      UU  PPPPPPPP
RRRRRRRR    00      00 LL         LL         UU      UU  PPPPPPPP
RR  RR      00      00 LL         LL         UU      UU  PP
RR  RR      00      00 LL         LL         UU      UU  PP
RR    RR    00      00 LL         LL         UU      UU  PP            ....
RR    RR    00      00 LL         LL         UU      UU  PP            ....
RR      RR    000000   LLLLLLLLLL LLLLLLLLLL UUUUUUUUUU  PP            ....
RR      RR    000000   LLLLLLLLLL LLLLLLLLLL UUUUUUUUUU  PP            ....

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
```

RO

AR

LA

FU

CO

CO

```
C
C Version:      'V04-000'
C
C****************************************************************************
C*                                                                          *
C*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                 *
C*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                  *
C*  ALL RIGHTS RESERVED.                                                    *
C*                                                                          *
C*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED   *
C*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE   *
C*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER   *
C*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY   *
C*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY   *
C*  TRANSFERRED.                                                            *
C*                                                                          *
C*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE   *
C*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT   *
C*  CORPORATION.                                                            *
C*                                                                          *
C*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS   *
C*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                 *
C*                                                                          *
C*                                                                          *
C****************************************************************************
C
C               AUTHOR  BRIAN PORTER            CREATION DATE   10-SEP-1980

C++
C       Functional description:
C
C       This module displays the device error roll-up report.
C
C       Modified by:
C
C       V03-004 SAR0257         Sharon A. Reynolds      25-Apr-1984
C               Widened the output field for the device names to
C               accomodate the node name being logged now.
C
C       V03-003 SAR0094         Sharon A. Reynolds,     20-Jun-1983
C               Changed the carriage control in the 'format' statements
C               for use with ERF.
C
C       V03-002 SAR009          Sharon Reynolds,        7-Apr-1983
C               Removed search_sid and entry_type as call parameters.
C               Added include statement for the 'msghdr' common which
C               defines the field definitions for the entry header
C               and equivalenced search_sid and entry_type to the
C               appropriate definitions.
C
C       v03-001 BP0004          Brian Porter,           05-APR-1982
C               Added stuff for mscp.
C**
C--

C++
C
```

```
C       +----------------------------------+
C       :             flink1               :
C       +----------------------------------+
C       :             blink1               :
C       +----------------------------------+
C       :           logging sid            :
C       +----------------------------------+
C       :         root name flink          :
C       +----------------------------------+
C       :         root name blink          :
C       +----------------------------------+
C       :        name entry count          :
C       +----------------------------------+
C
C
C       +----------------------------------+
C       :             flink2               :
C       +----------------------------------+
C       :             blink2               :
C       +----------------------------------+
C       :                                  :
C       +--                              --+
C       :         16 bytes for name        :
C       +--                              --+
C       :              string              :
C       +--                              --+
C       :                                  :
C       +----------------------------------+
C       :          root unit flink         :
C       +----------------------------------+
C       :          root unit blink         :
C       +----------------------------------+
C       :         unit entry count         :
C       +----------------------------------+
C
C
C       +----------------------------------+
C       :             flink3               :
C       +----------------------------------+
C       :             blink3               :
C       +----------------------------------+
C       :          ucb unit number         :
C       +----------------------------------+
C       :     device hard error count      :
C       +----------------------------------+
C       :     device soft error count      :
C       +----------------------------------+
C       :     timeout hard error count     :
C       +----------------------------------+
C       :     timeout soft error count     :
C       +----------------------------------+
C       :    ucb unit operation count      :
C       +----------------------------------+
C       :       ucb unit error count       :
C       +----------------------------------+
C       :            pack count            :
C       +----------------------------------+
C       :                                 :
C       +--                             --+
C       :          current label          :
C       +--        previous error       --+
C       :                                 :
C       +---------------------------------+
C--


        subroutine rollup (search_name_length,search_name_string,
        1 search_unit,iosb,ucb$l_opcnt,ucb$w_errcnt)

        Include 'SRC$:MSGHDR.FOR /NOLIST'       ! EMB entry header definitions

        byte            lun

        integer*4       buffer0(3)

        integer*4       root_logging_sid_flink

        integer*4       root_logging_sid_blink

        integer*4       logging_sid_entry_count

        Equivalence     (emb$l_hd_sid,search_sid)

        Equivalence     (emb$w_hd_entry,entry_type)

        equivalence     (buffer0(1),root_logging_sid_flink)

        equivalence     (buffer0(2),root_logging_sid_blink)

        equivalence     (buffer0(3),logging_sid_entry_count)

        integer*4       buffer1(6)

        integer*4       flink1

        integer*4       blink1

        integer*4       logging_sid

        integer*4       root_name_flink

        integer*4       root_name_blink

        integer*4       name_entry_count

        equivalence     (buffer1(1),flink1)

        equivalence     (buffer1(2),blink1)

        equivalence     (buffer1(3),logging_sid)

        equivalence     (buffer1(4),root_name_flink)
        equivalence     (buffer1(5),root_name_blink)

        equivalence     (buffer1(6),name_entry_count)

        integer*4       buffer2(9)

        integer*4       flink2

        integer*4       blink2

        byte            name_string_array(0:15)

        byte            name_string_length

        character*15    name_string

        equivalence     (name_string_array(0),name_string_length)

        equivalence     (name_string_array(1),name_string)

        integer*4       root_unit_flink

        integer*4       root_unit_blink

        integer*4       unit_entry_count

        equivalence     (buffer2(1),flink2)

        equivalence     (buffer2(2),blink2)

        equivalence     (buffer2(3),name_string_array)

        equivalence     (buffer2(7),root_unit_flink)

        equivalence     (buffer2(8),root_unit_blink)

        equivalence     (buffer2(9),unit_entry_count)

        integer*4       buffer3(13)

        integer*4       flink3

        integer*4       blink3

        integer*4       ucb_unit_number

        integer*4       device_hard_error_count

        integer*4       device_soft_error_count

        integer*4       timeout_hard_error_count

        integer*4       timeout_soft_error_count

        integer*4       ucb_unit_operation_count

        integer*4       ucb_unit_error_count

        integer*4       pack_count

        byte            previous_label_at_error_array(12)

        character*12    previous_label_at_error

        equivalence     (previous_label_at_error_array,previous_label_at_error)

        equivalence     (buffer3(1),flink3)

        equivalence     (buffer3(2),blink3)

        equivalence     (buffer3(3),ucb_unit_number)

        equivalence     (buffer3(4),device_hard_error_count)

        equivalence     (buffer3(5),device_soft_error_count)

        equivalence     (buffer3(6),timeout_hard_error_count)

        equivalence     (buffer3(7),timeout_soft_error_count)

        equivalence     (buffer3(8),ucb_unit_operation_count)

        equivalence     (buffer3(9),ucb_unit_error_count)

        equivalence     (buffer3(10),pack_count)

        equivalence     (buffer3(11),previous_label_at_error_array)

        integer*4       logging_sid_entry_address

        integer*4       name_entry_address

        integer*4       unit_entry_address

        character*15    search_name

        character*12    current_label_at_error

        logical*1       lib$extzv

        logical*1       lib$get_vm

        integer*4       search_sid

        byte            search_name_length

        character*15    search_name_string

        integer*2       search_unit

        integer*2       entry_type

        integer*4       compress4

        integer*4       column

        integer*4       iosb

        integer*4       ucb$l_opcnt

        integer*2       ucb$w_errcnt

        integer*4       device_count

        integer*4       insert_blink


        call movc5 (%val(search_name_length),%ref(search_name_string),%val(42),
        1 %val(15),%ref(search_name))

        logging_sid_entry_address = root_logging_sid_flink

        do 50,i = 1,logging_sid_entry_count

        call movc3 (%val(24),%val(logging_sid_entry_address),buffer1)

5       if (search_sid .eq. logging_sid) then

        name_entry_address = root_name_flink

        do 40,j = 1,name_entry_count

        call movc3 (%val(36),%val(name_entry_address),buffer2)

10      if (search_name .eq. name_string) then

        unit_entry_address = root_unit_flink

        do 30,k = 1,unit_entry_count

        call movc3 (%val(52),%val(unit_entry_address),buffer3)

15      if (search_unit .eq. ucb_unit_number) then

        if (
        1 ucb$l_opcnt .ne. -1
        1 .and.
        1 ucb$w_errcnt .ne. -1
        1 ) then

        if (
        1 ucb_unit_operation_count .eq. -1
        1 .and.
        1 ucb_unit_error_count .eq. -1
        1 ) then

        ucb_unit_operation_count = 0

        ucb_unit_error_count = 0
        endif

        ucb_unit_operation_count = ucb$l_opcnt

        ucb_unit_error_count = ucb$w_errcnt
        endif

        if (iosb .ne. -1) then

        if (
        1 entry_type .eq. 1
        1 .or.
        1 entry_type .eq. 98
        1 .or.
        1 entry_type .eq. 99
        1 .or.
        1 entry_type .eq. 100
        1 ) then

        if (lib$extzv(0,1,iosb) .eq. 1) then

        if (device_soft_error_count .eq. -1) then

        device_soft_error_count = 0
        endif

        device_soft_error_count = device_soft_error_count + 1
        else

        if (device_hard_error_count .eq. -1) then

        device_hard_error_count = 0
        endif

        device_hard_error_count = device_hard_error_count + 1
        endif

        else if (entry_type .eq. 96) then

        if (lib$extzv(0,1,iosb) .eq. 1) then

        if (timeout_soft_error_count .eq. -1) then

        timeout_soft_error_count = 0
        endif

        timeout_soft_error_count = timeout_soft_error_count + 1
        else

        if (timeout_hard_error_count .eq. -1) then

        timeout_hard_error_count = 0
        endif

        timeout_hard_error_count = timeout_hard_error_count + 1
        endif
        endif
        endif

        call get_current_label(3,search_sid,search_name_length,
        1 search_name_string,search_unit,%ref(current_label_at_error),*20)

        if (current_label_at_error .ne. previous_label_at_error) then

        pack_count = pack_count + 1

        previous_label_at_error = current_label_at_error
        endif

20      call movc3 (%val(44),ucb_unit_number,%val(unit_entry_address + 8))

        return
        endif

        insert_blink = blink3

        if (ucb_unit_number .gt. search_unit) goto 35

        unit_entry_address = flink3

30      continue

        insert_blink = root_unit_blink

35      call movc5 (%val(0),,%val(0),%val(52),buffer3)

        if (lib$get_vm(((52+7)/8)*8,unit_entry_address)) then

        call insque (%val(unit_entry_address),%val(insert_blink))

        ucb_unit_number = search_unit

        device_hard_error_count = -1

        device_soft_error_count = -1

        timeout_hard_error_count = -1

        timeout_soft_error_count = -1

        ucb_unit_operation_count = -1

        ucb_unit_error_count = -1

        unit_entry_count = unit_entry_count + 1

        call movl (unit_entry_count,%val(name_entry_address + 32))

        goto 15
        endif

        return
        endif
        name_entry_address = flink2

 40     continue

        call movc5 (%val(0),,%val(0),%val(36),buffer2)

        if (lib$get_vm(((36+7)/8)*8,name_entry_address)) then

        call insque (%val(name_entry_address),%val(root_name_blink))

        name_string_length = search_name_length

        name_string = search_name

        root_unit_flink = name_entry_address + 24

        root_unit_blink = root_unit_flink

        call movc3 (%val(28),name_string_length,%val(name_entry_address + 8))

        name_entry_count = name_entry_count + 1

        call movl (name_entry_count,%val(logging_sid_entry_address + 20))

        goto 10
        endif

        return
        endif

        logging_sid_entry_address = flink1

 50     continue

        call movc5 (%val(0),,%val(0),%val(24),buffer1)

        if (lib$get_vm(((24+7)/8)*8,logging_sid_entry_address)) then

        if (logging_sid_entry_count .eq. 0) then

        root_logging_sid_flink = %loc(root_logging_sid_flink)

        root_logging_sid_blink = root_logging_sid_flink
        endif

        call insque (%val(logging_sid_entry_address),
        1 %val(root_logging_sid_blink))

        logging_sid = search_sid

        root_name_flink = logging_sid_entry_address + 12

        root_name_blink = root_name_flink

        call movc3 (%val(16),logging_sid,%val(logging_sid_entry_address + 8))

        logging_sid_entry_count = logging_sid_entry_count + 1

        goto 5
        endif

        return



        entry display_rollup (lun)




        logging_sid_entry_address = root_logging_sid_flink

        do 100,i = 1,logging_sid_entry_count

        call movc3 (%val(24),%val(logging_sid_entry_address),buffer1)

        call frctof (lun)

        call linchk (lun,7)

        write(lun,55) logging_sid
55      format(/' ','DEVICE ROLLUP LOGGED BY SID ',z8.8,:a,/)

        write(lun,60)
60      format(/' ','DEVICE',t23,'ERROR BITS',t51,'ERRORS THIS',
        1 t64,'QIOS THIS',/,
        1 t26,'SET',t38,'QIO TIMEOUT',t53,'SESSION',t65,'SESSION')

        write(lun,65)
65      format(/' ',t21,'[HARD]',t28,'[SOFT]',
        1 t36,'[HARD]',t43,'[SOFT]')

        device_count = 0

        name_entry_address = root_name_flink

        do 90,j = 1,name_entry_count

        call movc3 (%val(36),%val(name_entry_address),buffer2)

        unit_entry_address = root_unit_flink

        do 80,k = 1,unit_entry_count

        if (device_count .eq. 23) then

        call frctof (lun)

        device_count = 0
        endif

        call movc3 (%val(44),%val(unit_entry_address),buffer3)

```

```
             call linchk (lun,2)

             write(lun,66)
    66       format(' ',:)

             write(lun,67) name_string,ucb_unit_number
    67       format(' ',' ',a<name_string_length>,
             1 i<compress4 (ucb_unit_number)>,':')

             column = name_string_length + (compress4 (ucb_unit_number)) + 2

             if (device_hard_error_count .EQ. -1) then
             Device_hard_error_count = 0
             Endif

             if (device_soft_error_count .EQ. -1) then
             Device_soft_error_count = 0
             Endif

             if (timeout_hard_error_count .EQ. -1) then
             Timeout_hard_error_count = 0
             Endif

             if (timeout_soft_error_count .EQ. -1) then
             Timeout_soft_error_count = 0
             Endif

             if (ucb_unit_error_count .EQ. -1) then
             Ucb_unit_error_count = 0
             Endif

             if (ucb_unit_operation_count .EQ. -1) then
             Ucb_unit_operation_count = 0
             Endif

             write(lun,69) device_hard_error_count,device_soft_error_count,
             1 timeout_hard_error_count,timeout_soft_error_count,
             2 ucb_unit_error_count,ucb_unit_operation_count

    69       format(T21,I5,'.',T28,I5,'.',
             1 T36,I5,'.',T43,I5,'.',
             2 T53,I6,'.',T64,I7,'.')

             device_count = device_count + 1

             unit_entry_address = flink3

    80       continue

             name_entry_address = flink2

    90       continue

             logging_sid_entry_address = flink1

    100      continue
```

```
                return

                end



PROGRAM SECTIONS

    Name                                        Bytes   Attributes

  0 $CODE                                        1477   PIC CON REL LCL   SHR   EXE   RD NOWRT LONG
  1 $PDATA                                        288   PIC CON REL LCL   SHR NOEXE   RD NOWRT LONG
  2 $LOCAL                                        588   PIC CON REL LCL NOSHR NOEXE   RD   WRT LONG
  3 EMB                                           512   PIC OVR REL GBL   SHR NOEXE   RD   WRT LONG

    Total Space Allocated                        2865

ENTRY POINTS

    Address  Type  Name                                  Address  Type  Name

  0-00000357       DISPLAY_ROLLUP                      0-00000000       ROLLUP

VARIABLES

    Address  Type  Name                                  Address  Type  Name

  2-0000005C  I*4  BLINK1                              2-00000038  I*4  BLINK2
  2-00000004  I*4  BLINK3                              2-000000A4  I*4  COLUMN
  2-0000008B  CHAR CURRENT_LABEL_AT_ERROR              2-000000A8  I*4  DEVICE_COUNT
  2-0000000C  I*4  DEVICE_HARD_ERROR_COUNT             2-00000010  I*4  DEVICE_SOFT_ERROR_COUNT
  3-00000000  I*4  EMB$L_HD_SID                        3-00000004  I*2  EMB$W_HD_ENTRY
  3-0000000E  I*2  EMB$W_HD_ERRSEQ                     3-00000004  I*2  ENTRY_TYPE
  2-00000058  I*4  FLINK1                              2-00000034  I*4  FLINK2
  2-00000000  I*4  FLINK3                              2-000000B0  I*4  I
  2-000000AC  I*4  INSERT_BLINK                       AP-00000010@ I*4  IOSB
  2-000000B4  I*4  J                                   2-000000B8  I*4  K
  2-00000060  I*4  LOGGING_SID                         2-00000098  I*4  LOGGING_SID_ENTRY_ADDRESS
  2-00000078  I*4  LOGGING_SID_ENTRY_COUNT            AP-00000004@ L*1  LUN
  2-0000009C  I*4  NAME_ENTRY_ADDRESS                  2-0000006C  I*4  NAME_ENTRY_COUNT
  2-0000003D  CHAR NAME_STRING                         2-0000003C  L*1  NAME_STRING_LENGTH
  2-00000024  I*4  PACK_COUNT                          2-00000028  CHAR PREVIOUS_LABEL_AT_ERROR
  2-00000074  I*4  ROOT_LOGGING_SID_BLINK              2-00000070  I*4  ROOT_LOGGING_SID_FLINK
  2-00000068  I*4  ROOT_NAME_BLINK                     2-00000064  I*4  ROOT_NAME_FLINK
  2-00000050  I*4  ROOT_UNIT_BLINK                     2-0000004C  I*4  ROOT_UNIT_FLINK
  2-0000007C  CHAR SEARCH_NAME                        AP-00000004@ L*1  SEARCH_NAME_LENGTH
 AP-00000008@ CHAR SEARCH_NAME_STRING                  3-00000000  I*4  SEARCH_SID
 AP-0000000C@ I*2  SEARCH_UNIT                         2-00000014  I*4  TIMEOUT_HARD_ERROR_COUNT
  2-00000018  I*4  TIMEOUT_SOFT_ERROR_COUNT           AP-00000014@ I*4  UCB$L_OPCNT
 AP-00000018@ I*2  UCB$W_ERRCNT                        2-00000020  I*4  UCB_UNIT_ERROR_COUNT
  2-00000008  I*4  UCB_UNIT_NUMBER                     2-0000001C  I*4  UCB_UNIT_OPERATION_COUNT
  2-000000A0  I*4  UNIT_ENTRY_ADDRESS                  2-00000054  I*4  UNIT_ENTRY_COUNT
```

```
ARRAYS

   Address  Type  Name                               Bytes  Dimensions

 2-00000070  I*4  BUFFER0                               12  (3)
 2-00000058  I*4  BUFFER1                               24  (6)
 2-00000034  I*4  BUFFER2                               36  (9)
 2-00000000  I*4  BUFFER3                               52  (13)
 3-00000000  L*1  EMB                                  512  (0:511)
 3-00000006  I*4  EMB$Q_HD_TIME                          8  (2)
 2-0000003C  L*1  NAME_STRING_ARRAY                     16  (0:15)
 2-00000028  L*1  PREVIOUS_LABEL_AT_ERROR_ARRAY         12  (12)


LABELS

   Address   Label       Address   Label       Address   Label       Address   Label       Address   Label       Address   Label

 0-0000004F  5        0-0000007C  10        0-000000AE  15        0-000001CB  20              **  30        0-00000204  35
       **    40             **    50        1-00000020  55'       1-00000049  60'       1-000000AD  65'       1-000000DA  66'
 1-000000DF  67'      1-000000F5  69'             **    80              **    90              **  100


FUNCTIONS AND SUBROUTINES REFERENCED

 Type  Name                         Type  Name                          Type  Name

  I*4  COMPRESS4                          FRCTOF                              GET_CURRENT_LABEL
       INSQUE                        L*1  LIBSEXTZV                     L*1  LIB$GET_VM
       LINCHK                             MOVC3                              MOVC5
       MOVL


COMMAND QUALIFIERS

  FORTRAN /LIS=LIS$:ROLLUP/OBJ=OBJ$:ROLLUP MSRC$:ROLLUP

  /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:          5.92 seconds
  Elapsed Time:      14.57 seconds
  Page Faults:       197
  Dynamic Memory:    218 pages
```

0153 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

RECSELECT
LIS

RLDISK
LIS

PUDRIVER
LIS

PCL11T
LIS

ROLLUP
LIS

RXDISK
LIS

PCL11R
LIS

SB11
LIS

RKDISK
LIS