

ISIS-11 PL/M-86 V2.0 COMPILATION OF MODULE VCMODE
 OBJECT MODULE PLACED IN VCMODE.OBJ
 COMPILER INVOKED BY: :F0: VCMODE.PLX XREF OPTIMIZE(3) DEBUG

```
$title("VCMODE.CMD - Set Virtual Console Background Mode")
```

```
$compact
```

```
vcmode:
```

```
do:
```

```
$include (:f2:copyrt.lit)
```

```
/*
```

```
Copyright (C) 1983
```

```
Digital Research
```

```
P.O. Box 579
```

```
Pacific Grove, CA 93950
```

```
*/
```

```
$include (:f2:vaxcmd.lit)
```

```
**** VAX commands for generation - read the name of this program  
for PRUGNAME below.
```

```
$ util := PROGNAME
```

```
$ ccpmsetup          I set up environment
```

```
$ assign "f$directory()" fl:          I use local dir for temp files
```

```
$ plm86 "util".plm xref "pl" optimize(3) debug
```

```
$ link86 f2:scd.obj, "util".obj to "util".lnk
```

```
$ loc86 "util".lnk od(sm(code,uats,data,stack,const)) -  
ad(sm(code(0),dats(10000h))) ss(stack(+32)) to "util".
```

```
$ h86 "util"
```

```
**** Then, on a micro:
```

```
A>vax progname.h86 $fans
```

```
A>gencmd progname data[b1000]
```

```
**** Notes: Stack is increased for interrupts. Const(ants) are last  
to force hex generation.
```

```
****/
```

```
$include (:f2:comlit.lit)
```

```
declare
```

lit	literally	"literally",
dcl	lit	"declare",
true	lit	"Offh",
false	lit	"0",
no	lit	"not",
boolean	lit	"byte",
forever	lit	"while true",
cr	lit	"13",
lf	lit	"10",
tab	lit	"9",

```
$include (:f2:mfunc.lit)
```

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

```

= /* Concurrent CP/M function numbers */
=
3 1 = dcl      m$prtbuff      lit      '9',
=          m$select          lit      '14',
=          m$openf           lit      '15',
=          m$closef          lit      '16',
=          m$deletef         lit      '19',
=          m$readf           lit      '20',
=          m$writef          lit      '21',
=          m$makef           lit      '22',
=          m$getlogin         lit      '24',
=          m$curisk           lit      '25',
=          m$setdma           lit      '26',
=          m$setatt           lit      '30',
=          m$setusr           lit      '32',
=          m$readrf           lit      '33',
=          m$writerf          lit      '34',
=          m$resetdrv         lit      '37',
=          m$errmode          lit      '45',
=          m$dirbios          lit      '50',
=          m$makeq            lit      '134',
=          m$openq            lit      '135',
=          m$deleteq          lit      '136',
=          m$readq            lit      '137',
=          m$creadq           lit      '138',
=          m$writeq           lit      '139',
=          m$cwriteq          lit      '140',
=          m$delay            lit      '141',
=          m$dispatch         lit      '142',
=          m$setprior         lit      '145',
=          m$attach           lit      '146',
=          m$detach           lit      '147',
=          m$setcns           lit      '148',
=          m$parse            lit      '152',
=          m$getcns           lit      '153',
=          m$sysdat           lit      '154',
=          m$jetpd            lit      '156',
=          m$abort            lit      '157';
=
= /* Internal calls */
=
4 1 = dcl      m$sleep         lit      '0212H',
=          m$wakeu           lit      '0213H';
=
= $include (:f2:fcblit)
=
5 1 = declare
=          f$drvusr           lit      '0',      /* drive/user byte */
=          f$name             lit      '1',      /* file name */
=          f$namelen          lit      '8',      /* file name length */
=          f$type             lit      '9',      /* file type field */
=          f$typelen          lit      '3',      /* type length */
=          f$rw               lit      '9',      /* high bit is R/W attribute */
=          f$dirsys           lit      '10',     /* high bit is dir/sys attribute */
=          f$ar               lit      '11',     /* high bit is archive attribute */
=          f$ex               lit      '12',     /* extent */
=          f$sl               lit      '13',     /* module byte */

```

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

```

=      f$rc          lit '15',      /* record count          */
=      f$diskmap     lit '16',      /* file disk map         */
=      diskmaplen    lit '16',      /* disk map length       */
=      f$drvusr2     lit '16',      /* fcb2                   */
=      f$name2       lit '17',
=      f$type2       lit '25',
=      f$cr          lit '32',      /* current record        */
=      f$rrc         lit '33',      /* random record         */
=      f$rrco        lit '35';      /* " " overflow         */
=

```

```

6 1      dcl fcb (1) byte external;
7 1      dcl buff (1) byte external;

      $include (:f2:sd.lit)
=
=      /* System Data Page */
=
8 1      dcl sysdat$pointer pointer;
9 1      dcl sysdat$ptr structure(
=      offset word,
=      segment word) at (@sysdat$pointer);
10 1     declare sd based sysdat$pointer structure (
=      supmod (4) word,
=      /* rtmmod (4) word,
=      memmod (4) word,
=      ciomod (4) word,
=      bdosmod (4) word,
=      xiosmod (4) word,
=      netmod (4) word,
=      reservd (4) word */
=      space(28) word,
=      mpmseg word,
=      rspseg word,
=      endseg word,
=      module$map byte,
=      ncns byte,
=      nlst byte,
=      nccb byte,
=      nflags byte,
=      srchdisk byte,
=      mmp word,
=      nslaves byte,
=      dayfile byte,
=      tempdisk byte,
=      tickspersec byte,
=      lul word,
=      ccb word,
=      flags word,
=      mdul word,
=      mfl word,
=      pul word,
=      qul word,
=      qmau (4) word,
=      rlr word,
=      dlr word,
=      drl word,
=

```

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

```

=      plr word,
=      slr word,
=      thrdrt word,
=      qlr word,
=      mal word,
=      version word,
=      vernum word,
=      mpavernum word,
=      tod (2) word,
=      tod_sec byte,
=      ncondev byte,
=      nlstdev byte,
=      nciodev byte,
=      lcb (2) word,
=      lckmax byte,
=      opmax byte,
=      systot (2) word,
=      cmod byte );
=
11  1  = declare sd$byte based sysdat$pointer (1) byte;
=
12  1  dcl ccb$pointer pointer;
13  1  dcl ccb$ptr structure ( offset address, segment address) at
      (@ccb$pointer);
      $include (:f2:vcch.lit)
      /* Concurrent CP/M Character Control Block Structure */
=
=  /*
=      00      |-----|-----|-----|
=              | attach      | queue      |
=              |-----|-----|-----|
=      04      | flag  | startcol | column  | nchar  |
=              |-----|-----|-----|
=      08      | mimic | source  | pc      | vc      |
=              |-----|-----|-----|
=      0C      | btmp  | rsvd    | state   |
=              |-----|-----|-----|
=      10      | maxbufsiz | vinq      |
=              |-----|-----|-----|
=      14      | voutq   | vcmxq   |
=              |-----|-----|-----|
=      18      | qpbflgs | qpbfill | qpbqaddr |
=              |-----|-----|-----|
=      1C      | qpbmsgs  | qpbbuffptr |
=              |-----|-----|-----|
=      20      | qbuff   | cosleep  |
=              |-----|-----|-----|
=      24      | usleep   | vsleep   |
=              |-----|-----|-----|
=      28      | ... reserved ... |
=              |-----|-----|-----|
=
=  */
=
=  */

```

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

```

14 1 = dcl ccb$structure lit 'structure (attach address, queue address,
    = flag byte, startcol byte, column byte, nchar byte, mimic byte, msource byte,
    = ccb$tail1';
15 1 = dcl ccb$tail1 lit
    = 'pc byte, vc byte, btmp byte, rsvd byte, state word, maxbufsiz word,
    = ccb$tail2';
16 1 = dcl ccb$tail2 lit
    = 'vinq address, voutq address, vcmxq address,
    = qpbflags byte, qpbresrvd byte, qpbqaddr address,
    = qpbmsgs address, qpbbuffptr address, qbuff address, cosleep word,
    = usleep word, vsleep word, r1 word, r2 word);
17 1 = declare
    = cfb$listcp      lit      '001h', /* flag values          */
    = cfb$compc      lit      '002h', /* control P toggle     */
    = cfb$switchs    lit      '004h', /* suppress output      */
    = cfb$conout      lit      '008h', /* XIOS supports switch */
    = cfb$vout        lit      '010h', /* XIOS console output  */
    = /* values of state byte */
    = /* conout goes to XIOS */
    = /* state word flags */
18 1 = dcl
    = csm$buffered    lit      '0001h',
    = csm$background  lit      '0002h',
    = csm$purging      lit      '0004h',
    = csm$noswitch     lit      '0008h',
    = csm$suspend      lit      '0010h',
    = csm$abort        lit      '0020h',
    = csm$filefull     lit      '0040h',
    = csm$ctrls        lit      '0080h',
    = csm$ctrl0        lit      '0100h',
    = csm$ctrlp        lit      '0200h';
19 1 = dcl x$init$offset lit '0Ch',
    = x$init$pointer pointer,
    = x$init$ptr structure (offset word, segment word) at (@x$init$pointer),
    = x$init based x$init$pointer structure
    = (tick byte, ticks$sec byte, door byte, resrvd1 (2) byte,
    = nvcns byte, nccb byte, nlst byte, ccb word, lcb word);
20 1 = dcl lcb$structure lit 'structure (attach address, queue address,
    = flag byte, startcol byte, column byte, nchar byte,
    = mimic byte, msource byte);
21 1 = dcl ccb based ccb$pointer ccb$structure;
22 1 = dcl ccpmproduct lit '14h';
23 1 = dcl bdosversion lit '31h';
24 1 = mon1: procedure (func,info) external;
25 2 = declare func byte;
26 2 = declare info address;

```

COPYRIGHT © 1981, 1982, 1983
 DIGITAL RESEARCH
 P. O. BOX 579
 PACIFIC GROVE, CA 93950

SER. # _____

```

27  2      end mon1;

28  1      mon2: procedure (func,info) byte external;
29  2          declare func byte;
30  2          declare info address;
31  2      end mon2;

32  1      mon3: procedure (func,info) address external;
33  2          decl func byte, info address;
34  2      end mon3;

35  1      mon4: procedure (func,info) pointer external;
36  2          decl func byte, info address;
37  2      end mon4;

```

```

/*****
*
*      B D O S      Externals
*
*****/

```

```

38  1      print$char: procedure(char);
39  2          declare char byte;
40  2          call mon1(2,char);
41  2      end print$char;

42  1      print$console$buffer: procedure (buffer$address);
43  2          declare buffer$address address;
44  2          call mon1 (9,buffer$address);
45  2      end print$console$buffer;

46  1      version: procedure address;
47  2          return mon3(12,0);
48  2      end version;

```

```

/*****
*
*      X D O S      Externals
*
*****/

```

```

49  1      terminate: procedure;
50  2          call mon1 (143,0);
51  2      end terminate;

52  1      get$console$number: procedure byte;
53  2          return mon2 (153,0);
54  2      end get$console$number;

55  1      printb: procedure public;
56  2          call print$char(' ');
57  2      end printb;

58  1      pdecimal: procedure(v,prec,zerosup) public;

```

```

/* print value v, field size = (log10 prec) + 1 */

```

COPYRIGHT © 1981, 1982, 1983
 DIGITAL RESEARCH
 P. O. BOX 579
 PACIFIC GROVE, CA 93950

SER. # _____

```

59  2      declare v address,      /* with leading zero suppression if zerosup = true */
        prec address,             /* value to print */
        zerosup boolean,          /* precision */
        d byte;                   /* zero suppression flag */
                                   /* current decimal digit */

60  2      do while prec <> 0;
61  3          d = v / prec;         /* get next digit */
62  3          v = v mod prec;       /* get remainder back to v */
63  3          prec = prec / 10;     /* ready for next digit */
64  3          if prec <> 0 and zerosup and d = 0 then
65  3              call printb;
        else
66  3              do;
67  4                  zerosup = false;
68  4                  call printchar("0"+d);
69  4              end;
70  3          end;
71  2      end pdecimal;

/*lbracket: procedure byte; /* find left bracket in command tail return */
/* dcl i byte;             /* its index. if not found ret 0 */
/* i = 1;
do while i <= buff(0) and (buff(i) = " " or buff(i) = tab);
i = i + 1;
end;
if buff(i) = "L" then
return(i);
return(0);
end lbracket;
*/

72  1      help: procedure;
73  2          call monl(m$prt$buf, .(cr, lf, tab, tab, tab, "VCMODE EXAMPLES$"));
74  2          call monl(m$prt$buf, .(cr, lf, lf, "vcmode", tab, tab, tab, tab, tab,
        "(show background mode)$"));
75  2          call monl(m$prt$buf, .(cr, lf, "vcmode dynamic", tab, tab, tab, tab,
        "(sets background mode)$"));
76  2          call monl(m$prt$buf, .(cr, lf, "vcmode buffered", tab, tab, tab, tab,
        tab, "$"));
        /* call monl(m$prt$buf, .(cr, lf, "vcmode suspend", tab, tab, tab, tab,
        tab, "$")); */
77  2          call monl(m$prt$buf, .(cr, lf, "vcmode size = 5", tab, tab, tab, tab,
        "(sets buffered mode max file size in)$"));
78  2          call monl(m$prt$buf, .(cr, lf, tab, tab, tab, tab, tab,
        "(kilobytes, legal range is 1 to 8191)$"));
79  2          call monl(m$prt$buf, .(cr, lf, tab, tab, tab, tab, tab,
        "(also sets background mode to buffered)$"));
        /* call monl(m$prt$buf, .(cr, lf, "vcmode size = 100H", tab, tab, tab,
        "(legal range in HEX is 1H to 1FFFFH)$")); */
80  2          call monl(m$prt$buf, .(cr, lf, "vcmode help", tab, tab, tab, tab,
        "(prints this message)$"));
81  2          call monl(m$prt$buf, .(cr, lf, "$"));
82  2          call terminate;
83  2      end help;

84  1      showstate: procedure (verb);

```

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

```

85 2      dcl (verb,state) address;
86 2      call monl(m$prt$buf,.(cr,lf,"background Mode For Virtual Console$"));
87 2      call pdecimal (console, 100, true);
88 2      call printb;
89 2      call monl(m$prt$buf, verb);
90 2      state = ccb.state and csm$buffered;
91 2      if state = 0 then
92 2          call monl(m$prt$buf,.( " Dynamic$"));
          else
93 2          do;
94 3              call monl(m$prt$buf,.( " Buffered", cr, lf, "Maximum file size = $"));
95 3              call pdecimal(ccb.maxbufsiz, 10000, true);
96 3              call monl(m$prt$buf,.( "K$"));
97 3          end;
98 2      call monl(m$prt$buf,.(cr, lf, "$"));

```

```

99 2      end show$state;

```

```

      $include (if2:qd.lit)

```

```

      /* Queue Descriptor */

```

```

100 1      dcl qnamsiz lit "8";

```

```

101 1      dcl qd$structure lit "structure(
      link word,
      net byte,
      org byte,
      flags word,
      name(qnamsiz) byte,
      msglen word,
      nmsgs word,
      dq word,
      nq word,
      msgcnt word,
      msgout word,
      buffer word)";

```

```

      /* queue flag values */

```

```

102 1      dcl qf$mx      lit "001h"; /* Mutual Exclusion */
103 1      dcl qf$keep    lit "002h"; /* NO DELETE      */
104 1      dcl qf$hide    lit "004h"; /* Not User writable */
105 1      dcl qf$rsp     lit "008h"; /* rsp queue      */
106 1      dcl qf$stable  lit "010h"; /* from qd table  */
107 1      dcl qf$rpl     lit "020h"; /* rpl queue      */
108 1      dcl qf$dev     lit "040h"; /* device queue   */

```

```

      /* Queue Parameter Block */

```

```

109 1      dcl qpb$structure lit "structure(
      flgs byte,
      net byte,
      qaddr word,
      nmsgs word,
      buffptr word,
      name (qnamsiz) byte )";

```

COPYRIGHT © 1981, 1982, 1983
 DIGITAL RESEARCH
 P. O. BOX 579
 PACIFIC GROVE, CA 93950

SER. # _____

COPYRIGHT © 1981, 1982, 1983
 DIGITAL RESEARCH
 P. O. BOX 579
 PACIFIC GROVE, CA 93950

SER. # _____

```

110 1      dcl qpb qpb$structure;
111 1      read$change$mxq: procedure;
112 2          qpb.qaddr = ccb.vcmxq;
113 2          call monl (m$readq, .qpb);
114 2      end read$change$mxq;

115 1      write$change$mxq: procedure;
116 2          qpb.qaddr = ccb.vcmxq;
117 2          call monl (m$writeq, .qpb);
118 2      end write$change$mxq;

119 1      atohb: procedure (char) byte public; /* convert ascii hex to nibble value */
120 2          declare char byte;
121 2          if char >= "0" and char <= "9" then
122 2              char = char - "0";
123 2          else if char >= "A" and char <= "F" then
124 2              char = char - "A" + 10;
125 2          else
126 2              char = 255;
127 2          return(char);
127 2      end atohb;

128 1      atodb: procedure (char) byte public; /* convert ascii decimal to nibble value */
129 2          declare char byte;
130 2          if char >= "0" and char <= "9" then
131 2              char = char - "0";
132 2          else
133 2              char = 255;
134 2          return(char);
134 2      end atodb;

135 1      atoi: procedure(str) word; /* convert ascii to 16 bit unsigned value */
136 2          dcl str pointer;
137 2          dcl (accum, temp) word;
138 2          dcl (val, i, len) byte;
139 2          dcl string based str (1) byte;
140 2          i, accum = 0;
141 2          if (len := findb(str, "H", 5)) <> 0ffffh then /* hex conversion */
142 2              do while (val := atohb(string(i))) <> 0ffh and i < len;
143 3                  accum = shl(accum, 4) + val;
144 3                  i = i + 1;
145 3              end;
146 2          else /* decimal is default base */
147 3              do while (val := atodb(string(i))) <> 0ffh and i < 5;
148 3                  accum = 10 * accum + val;
149 3                  if i = 4 then
150 3                      temp = accum;
151 3                      i = i + 1;
152 3                  end;
153 2          if temp > accum then /* overflow */
154 2              accum = 0ffffh;
155 2          return(accum);
155 2      end atoi;

```

```

156 1  compare: procedure(ustr, ostr, minlen, maxlen) boolean;
157 2  dcl (ustr, ostr) pointer; /* user string, option string */
158 2  dcl user$string based ustr (1) byte;
159 2  dcl (minlen, maxlen) byte;
160 2  dcl cmplen word;
161 2  cmplen = cmpb(ustr, ostr, maxlen);
162 2  if cmplen = 0ffffh or (user$string(cmplen) = ' ' and cmplen >= minlen) then
163 2  return(true);
164 2  if user$string(cmplen) = ' ' then
165 2  do;
166 3  call mon1(m$prt$buf, .(cr,lf,"Invalid Command Option.", cr ,lf, "$"));
167 3  call help;
168 3  end;
169 2  return(false);
170 2  end compare;

171 1  dcl vers address initial (0);
172 1  dcl no$state lit "0ffh";
173 1  dcl console byte;

/*
Main Program
*/

174 1  plmstart: procedure public;
175 2  dcl option$ptr byte;
176 2  dcl num word;
177 2  vers = version;
178 2  if (high(vers) <> ccpmproduct) then
179 2  do;
180 3  call print$console$buffer(.(cr,lf,"Requires Concurrent CP/M-86", cr, lf,
181 3  "$"));
182 3  call mon1(0,0);
183 3  end;

183 2  sysdat$pointer, ccb$pointer = mon4(m$sysdat, 0); /* system data segment */
184 2  ccb$ptr.offset = sd.ccb + (console := mon2(m$getchs, 0)) * size(ccb);

185 2  call read$change$mxq; /* MXQ is written in kernel terminate code */
186 2  if (ccb.state and csm$background) <> 0 then
187 2  call mon1(m$prt$buf, .(cr,lf,"Virtual Console not in foreground", cr, lf,
188 2  "$"));
188 2  else if buff(0) = 0 then
189 2  call show$state(.( "is$")); /* show current state */
190 2  else /* try to set state or show help message */
191 2  do;
192 3  fcb(f$type) = ' ';
193 3  if compare(@fcb(f$name), @("BUFFERED"), 1, 8) then
194 3  ccb.state = ccb.state or csm$buffered;
195 3  else if compare(@fcb(f$name), @("DYNAMIC"), 1, 7) then
196 3  ccb.state = ccb.state and not double(csm$buffered);
197 3  else if compare(@fcb(f$name), @("HELP"), 1, 4) then
198 3  call help;
199 3  else if compare(@fcb(f$name), @("SIZE"), 1, 4) then /* change to 2,4 */
200 3  do; /* when suspend is put back in */
201 4  num = atoi(@fcb(f$name2));
if num > 0 and num < 2000H then

```

COPYRIGHT © 1981, 1982, 1983
 DIGITAL RESEARCH
 P.O. BOX 579
 PACIFIC GROVE, CA 93950

SER. # _____

```
202 4      ccb.maxbufsiz = num;      /* limit size to 16 bit record count */
      else
203 4      do;
204 5          call mon1(m$prt$buf,.(cr,lf,"File size out of range", cr ,lf, "$"));
205 5          call help;
206 5      end;
207 4      ccb.state = ccb.state or csm$buffered;
      /* automatically sets to buffered */
208 4      end;
      else
209 3      do;
210 4          call mon1(m$prt$buf,.(cr,lf,"Invalid Command Option.", cr ,lf, "$"));
211 4          call help;
212 4      end;
213 3      call show$state(.( "set to$"));
214 3      end;
215 2      call terminate;

216 2      end plmstart;
217 1      end vcmode;
```

COPYRIGHT © 1981, 1982, 1983
DIGITAL RESEARCH
P. O. BOX 570
PACIFIC GROVE, CA 93950
SER. # _____

CROSS-REFERENCE LISTING

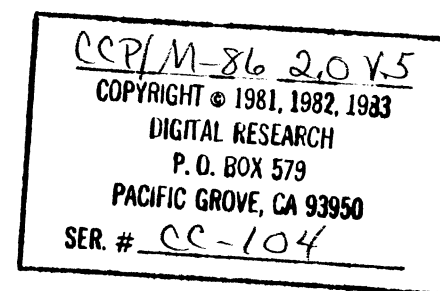
 DEFN ADDR SIZE NAME, ATTRIBUTES, AND REFERENCES

137	001EH	2	ACCUM.	WORD	140	143	147	149	152	153	154						
128	01F2H	32	ATODB.	PROCEDURE BYTE PUBLIC STACK=0004H							146						
119	01BEH	52	ATUHB.	PROCEDURE BYTE PUBLIC STACK=0004H							142						
135	0212H	215	ATOT.	PROCEDURE WORD STACK=000CH						200							
21	0000H	2	ATTACH.	WORD MEMBER(CCB)													
23			BOOSVERSION. . . .	LITERALLY													
2			BOOLEAN.	LITERALLY	59	156											
21	000CH	1	BTMP.	BYTE MEMBER(CCB)													
7	0000H	1	BUFF.	BYTE ARRAY(1) EXTERNAL(1)					188								
42	0004H	2	BUFFERADDRESS. . .	WORD PARAMETER AUTOMATIC					43	44							
110	0006H	2	BUFFPTR.	WORD MEMBER(QP3)													
19	0008H	2	CCB.	WORD MEMBER(XINIT)													
21	0000H	44	CCB.	STRUCTURE BASED(CCBPOINTER)						90	95	112	116	184	186	193	
				195 202 207													
10	0054H	2	CCB.	WORD MEMBER(SD)					184								
12	0004H	4	CCBPOINTER. . . .	POINTER	13	21	90	95	112	116	183	185	193	195	202		
				207													
13	0004H	4	CCBPTR.	STRUCTURE AT					184								
14			CCBSTRUCTURE. . . .	LITERALLY		21											
15			CCBTAIL1.	LITERALLY		21											
16			CCBTAIL2.	LITERALLY		21											
22			CCPMPRODUCT. . . .	LITERALLY		178											
17			CFCMPC.	LITERALLY													
17			CFCNDOUT.	LITERALLY													
17			CFLISTCP.	LITERALLY													
17			CFSWITCHS.	LITERALLY													
17			CFVOUT.	LITERALLY													
128	0004H	1	CHAR.	BYTE PARAMETER AUTOMATIC					129	130	131	132	133				
119	0004H	1	CHAR.	BYTE PARAMETER AUTOMATIC					120	121	122	123	124	125	126		
38	0004H	1	CHAR.	BYTE PARAMETER AUTOMATIC					39	40							
10	0090H	1	CMOD.	BYTE MEMBER(SD)													
			CMPB.	BUILTIN		161											
160	0022H	2	CMPLEN.	WORD		161	162	164									
21	0006H	1	COLUMN.	BYTE MEMBER(CCB)													
156	02E9H	111	COMPARE.	PROCEDURE BYTE STACK=001CH						192	194	196	198				
173	002CH	1	CONSOLE.	BYTE		87	184										
21	0022H	2	COSLEEP.	WORD MEMBER(CCB)													
2			CR.	LITERALLY		73	74	75	76	77	78	79	80	81	86	94	
				98 166 180 187 204 210													
18			CSHABORT.	LITERALLY													
18			CSMBACKGROUND. . . .	LITERALLY		186											
18			CSMBUFFERED. . . .	LITERALLY		90	193	195	207								
18			CSMCTRLD.	LITERALLY													
18			CSMCTRLP.	LITERALLY													
18			CSMCTRLS.	LITERALLY													
18			CSMFILEFULL. . . .	LITERALLY													
18			CSMNOSWITCH. . . .	LITERALLY													
18			CSMPURGING.	LITERALLY													
18			CSMSUSPEND.	LITERALLY													

COPYRIGHT © 1981, 1982, 1983
 DIGITAL RESEARCH
 P. O. BOX 579
 PACIFIC GROVE, CA 93950

SER. # _____

59	0028H	1	D.	BYTE	61	64	68												
10	004FH	1	DAYFILE.	BYTE MEMBER(SD)															
2			DCL.	LITERALLY															
5			DISKMAPLEN.	LITERALLY															
10	006AH	2	DLR.	WORD MEMBER(SD)															
19	0002H	1	DOOR.	BYTE MEMBER(XINIT)															
			DOUBLE.	BUILTIN	195														
10	006CH	2	DRL.	WORD MEMBER(SD)															
10	0044H	2	ENDSEG.	WORD MEMBER(SD)															
2			FALSE.	LITERALLY	67	169													
5			FARC.	LITERALLY															
6	0000H	1	FCB.	BYTE ARRAY(1) EXTERNAL(0)	191	192	194	196	198	200									
5			FCR.	LITERALLY															
5			FDIRSYS.	LITERALLY															
5			FDISKMAP.	LITERALLY															
5			FDRVUSR.	LITERALLY															
5			FDRVUSR2.	LITERALLY															
5			FEX.	LITERALLY															
			FINDB.	BUILTIN	141														
21	0004H	1	FLAG.	BYTE MEMBER(CCB)															
10	0056H	2	FLAGS.	WORD MEMBER(SD)															
110	0000H	1	FLGS.	BYTE MEMBER(CPB)															
5			FNAME.	LITERALLY	192	194	196	198											
5			FNAME2.	LITERALLY	200														
5			FNAMELEN.	LITERALLY															
2			FOREVER.	LITERALLY															
5			FRC.	LITERALLY															
5			FRREC.	LITERALLY															
5			FRRECO.	LITERALLY															
5			FRW.	LITERALLY															
5			FS1.	LITERALLY															
5			FTYPE.	LITERALLY	191														
5			FTYPE2.	LITERALLY															
5			FTYPELEN.	LITERALLY															
24	0000H	1	FUNC.	BYTE PARAMETER	25														
35	0000H	1	FUNC.	BYTE PARAMETER	36														
32	0000H	1	FUNC.	BYTE PARAMETER	33														
28	0000H	1	FUNC.	BYTE PARAMETER	29														
52	0043H	15	GETCONSOLENUMBER.	PROCEDURE BYTE STACK=0008H															
72	00B3H	98	HELP.	PROCEDURE STACK=000CH	167	197	205	211											
			HIGH.	BUILTIN	178														
138	0024H	1	I.	BYTE	140	142	144	146	148	150									
24	0000H	2	INFO.	WORD PARAMETER	26														
28	0000H	2	INFO.	WORD PARAMETER	30														
35	0000H	2	INFO.	WORD PARAMETER	36														
32	0000H	2	INFO.	WORD PARAMETER	33														
10	0086H	4	LCB.	WORD ARRAY(2) MEMBER(SD)															
19	000AH	2	LCB.	WORD MEMBER(XINIT)															
20			LCBSTRUCTURE.	LITERALLY															
10	008AH	1	LCKMAX.	BYTE MEMBER(SD)															
138	002BH	1	LEN.	BYTE	141	142													
2			LF.	LITERALLY	73	74	75	76	77	78	79	80	81	86	94				
					98	166	180	187	204	210									
2			LIT.	LITERALLY	2	3	4	5	14	15	16	17	19	19	20				
					22	23	100	101	102	103	104	105	106	107	108	109	172		
10	0052H	2	LUL.	WORD MEMBER(SD)															
3			MAORT.	LITERALLY															



10	0076H	2	MAL.	WORD MEMBER(SD)															
3			MATTACH.	LITERALLY															
21	0010H	2	MAXBUFSTZ.	WORD MEMBER(CCB)	95	202													
156	0004H	1	MAXLEN.	BYTE PARAMETER AUTOMATIC		159	161												
3			MCLOSEF.	LITERALLY															
3			MCREADQ.	LITERALLY															
3			MCURDSK.	LITERALLY															
3			MCWRITLQ.	LITERALLY															
3			MDELAY.	LITERALLY															
3			MDELETF.	LITERALLY															
3			MDELETF.	LITERALLY															
3			MDETACH.	LITERALLY															
3			MDIRBIDS.	LITERALLY															
3			MDISPATCH.	LITERALLY															
10	0058H	2	MOUL.	WORD MEMBER(CSD)															
3			MEKRMODE.	LITERALLY															
10	005AH	2	MFL.	WORD MEMBER(CSD)															
3			MGETCNS.	LITERALLY	184														
3			MGETLOGIN.	LITERALLY															
3			MGETPD.	LITERALLY															
21	0008H	1	MIMIC.	BYTE MEMBER(CCB)															
156	0006H	1	MINLEN.	BYTE PARAMETER AUTOMATIC		159	162												
4			MISLEEP.	LITERALLY															
4			MWAKEUP.	LITERALLY															
3			MMAKEF.	LITERALLY															
3			MMAKEQ.	LITERALLY															
10	004CH	2	MMP.	WORD MEMBER(CSD)															
10	0046H	1	MODULEMAP.	BYTE MEMBER(CSD)															
24	0000H		MON1.	PROCEDURE EXTERNAL(2) STACK=0000H		40	44	50	73	74	75								
				76 77 78 79 80 81 86 89 92 94 96 98 113 117															
				166 181 187 204 210															
28	0000H		MON2.	PROCEDURE BYTE EXTERNAL(3) STACK=0000H		53	184												
32	0000H		MON3.	PROCEDURE WORD EXTERNAL(4) STACK=0000H		47													
35	0000H		MON4.	PROCEDURE POINTER EXTERNAL(5) STACK=0000H			183												
3			MOPENF.	LITERALLY															
3			MOPENQ.	LITERALLY															
3			MPARSE.	LITERALLY															
10	0040H	2	MPMSEG.	WORD MEMBER(CSD)															
10	007CH	2	MPMVERNUM.	WORD MEMBER(CSD)															
3			MPRTBUF.	LITERALLY		73	74	75	76	77	78	79	80	81	86	89			
				92 94 96 98 166 187 204 210															
3			MREADF.	LITERALLY															
3			MREADQ.	LITERALLY	113														
3			MREADRF.	LITERALLY															
3			MRESETDRV.	LITERALLY															
3			MSELECT.	LITERALLY															
3			MSETATT.	LITERALLY															
3			MSETCNS.	LITERALLY															
3			MSETDMA.	LITERALLY															
3			MSETPRIOR.	LITERALLY															
3			MSETUSR.	LITERALLY															
21	0009H	1	MSOURCE.	BYTE MEMBER(CCB)															
3			MSYSDAT.	LITERALLY	183														
3			MWRITEF.	LITERALLY															
3			MWRITEQ.	LITERALLY	117														
3			MWRITERF.	LITERALLY															
110	0008H	8	NAME.	BYTE ARRAY(8) MEMBER(CPB)															

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

COPYRIGHT © 1981, 1982, 1983
DIGITAL RESEARCH
P. O. BOX 579
PACIFIC GROVE, CA 93950
SER. # _____

19	0003H	2	RESRVD1.	BYTE ARRAY(2) MEMBER(XINIT)															
10	0068H	2	RLR.	WORD MEMBER(SD)															
10	0042H	2	RSPSEG	WORD MEMBER(SD)															
21	0000H	1	RSVD	BYTE MEMBER(CCB)															
10	0000H	145	SD	STRUCTURE BASED(SYSDATPTR)															184
11	0000H	1	SD BYTE	BYTE BASED(SYSDATPTR) ARRAY(1)															
19	0002H	2	SEGMENT.	WORD MEMBER(XINITPTR)															
13	0002H	2	SEGMENT.	WORD MEMBER(CCBPTR)															
9	0002H	2	SEGMENT.	WORD MEMBER(SYSDATPTR)															
			SHL.	BUILTIN															143
84	0115H	117	SHOWSTATE.	PROCEDURE STACK=001EH															189 213
			SIZE	BUILTIN															184
10	0070H	2	SLR.	WORD MEMBER(SD)															
10	0008H	56	SPACE.	WORD ARRAY(28) MEMBER(SD)															
10	004BH	1	SRCHDTSK	BYTE MEMBER(SD)															
21	0005H	1	STARTCOL	BYTE MEMBER(CCB)															
21	000EH	2	STATE.	WORD MEMBER(CCB)															
85	000CH	2	STATE.	WORD															
																			90 91
135	0004H	4	STK.	POINTER PARAMETER AUTOMATIC															
139	0000H	1	STRING	BYTE BASED(STR) ARRAY(1)															
10	0000H	8	SUPMOD	WORD ARRAY(4) MEMBER(SD)															
8	0000H	4	SYSDATPTR.	POINTER															
																			9 10 11 183 184
9	0000H	4	SYSDATPTR.	STRUCTURE AT															
10	008CH	4	SYSLTOT.	WORD ARRAY(2) MEMBER(SD)															
2			TAB.	LITERALLY															
																			73 74 75 76 77 78 79 80
137	0020H	2	TEMP	WORD															
																			149 152
10	0050H	1	TERMDISK	BYTE MEMBER(SD)															
49	0034H	15	TERMINATE.	PROCEDURE STACK=0008H															
10	0072H	2	THDRT	WORD MEMBER(SD)															
19	0000H	1	TICK	BYTE MEMBER(XINIT)															
10	0051H	1	TICKSPERSEC.	BYTE MEMBER(SD)															
19	0001H	1	TICKSSEC	BYTE MEMBER(XINIT)															
10	007EH	4	TOD.	WORD ARRAY(2) MEMBER(SD)															
10	0082H	1	TOD_SEC.	BYTE MEMBER(SD)															
2			TRUE	LITERALLY															
																			87 95 163
158	0000H	1	USERSTRING	BYTE BASED(ISTR) ARRAY(1)															
21	0024H	2	USLEEP	WORD MEMBER(CCB)															
156	000CH	4	USTR	POINTER PARAMETER AUTOMATIC															
58	0008H	2	V.	WORD PARAMETER AUTOMATIC															
138	0029H	1	VAL.	BYTE															
																			142 143 146 147
21	000BH	1	VC	BYTE MEMBER(CCB)															
1	0002H		VCMODE	PROCEDURE STACK=0000H															
21	0016H	2	VCMXQ.	WORD MEMBER(CCB)															
																			112 116
84	0004H	2	VERB	WORD PARAMETER AUTOMATIC															
10	007AH	2	VERNUM	WORD MEMBER(SD)															
171	0024H	2	VERS	WORD INITIAL															
																			177 178
10	0078H	2	VERSION.	WORD MEMBER(SD)															
46	0025H	15	VERSION.	PROCEDURE WORD STACK=0008H															
																			177
21	0012H	2	VINQ	WORD MEMBER(CCB)															
21	0014H	2	VOUTQ.	WORD MEMBER(CCB)															
21	0026H	2	VSLEEP	WORD MEMBER(CCB)															
115	01A4H	26	WRITECHANGEMAX	PROCEDURE STACK=0008H															
19	0000H	12	XINIT.	STRUCTURE BASED(XINITPTR)															
19			XINTTOFFSET.	LITERALLY															
19	0008H	4	XINITPTR.	POINTER															
																			19
19	0008H	4	XINITPTR.	STRUCTURE AT															
58	0004H	1	ZEROSUP.	BYTE PARAMETER AUTOMATIC															
																			59 64 67

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. #

MODULE INFORMATION:

CODE AREA SIZE	= 04A8H	11920
CONSTANT AREA SIZE	= 024CH	5880
VARIABLE AREA SIZE	= 002FH	460
MAXIMUM STACK SIZE	= 0022H	340
584 LINES READ		
0 PROGRAM ERROR(S)		

END OF PL/M-86 COMPILATION

COPYRIGHT © 1981, 1982, 1983
DIGITAL RESEARCH
P. O. BOX 579
PACIFIC GROVE, CA 93950

SER. # _____



ISIS-II MCS-86 LOCATER, V1.2 INVOKED BY:

:F0: VCMODE.LNK ODC(SHCODE,DATS,DATA,STACK,CURST)) ADC(SHCODE(D),DATS(10000H)) SS(STACK(+32)) TO VCMODE.

SYMBOL TABLE OF MODULE SCD
READ FROM FILE VCMODE.LNK
WRITTEN TO FILE VCMODE.

BASE	OFFSET	TYPE	SYMBOL
------	--------	------	--------

0000H	0458H	PUB	PLMSTART
0000H	02BEH	PUB	ATONB
0000H	0152H	PUB	PRINTB
0000H	0050H	PUB	MON1
0000H	0050H	PUB	MON3
1000H	0004H	PUB	BDTSK
1000H	0050H	PUB	CADRK
1000H	0053H	PUB	LEN0
1000H	0056H	PUB	LEN1
1000H	006CH	PUB	FC916
1000H	007DH	PUB	RR
1000H	0080H	PUB	BUFF
1000H	0080H	PUB	BUFFA
1000H	0100H	PUB	SAVEAX
1000H	0104H	PUB	SAVECX

VCMODE: SYMBOLS AND LINES

1000H	03D0H	SYM	MEMORY
1000H	0108H	SYM	SYSDATPTR
1000H	0108H	BAS	SDBYTE
1000H	010CH	SYM	CCBPTR
1000H	0110H	SYM	XINITPTR
1000H	010CH	BAS	CCB
STACK	0004H	SYM	CHAR

STACK	0004H	SYM	BUFFERADDRESS
0000H	0134H	SYM	TERMTNATE
0000H	0152H	SYM	PRINTB
STACK	0008H	SYM	V
STACK	0004H	SYM	ZEROSUP
0000H	01B3H	SYM	HELP
STACK	0004H	SYM	VERB
1000H	0116H	SYM	QPB
0000H	02A4H	SYM	WRITECHANGEMXQ
STACK	0004H	SYM	CHAR
STACK	0004H	SYM	CHAR
STACK	0004H	SYM	STR
1000H	0128H	SYM	TEMP
1000H	0132H	SYM	I
STACK	0004H	BAS	STRING
STACK	000CH	SYM	USTR
STACK	0006H	SYM	MINLEN
STACK	000CH	BAS	USERSTRING
1000H	012CH	SYM	VERS
0000H	0458H	SYM	PLMSTART
1000H	012EH	SYM	NUM
0000H	0105H	LIN	40
0000H	0115H	LIN	42
0000H	0121H	LIN	45
0000H	0128H	LIN	47
0000H	0134H	LIN	49

BASE	OFFSET	TYPE	SYMBOL
------	--------	------	--------

0000H	02F2H	PUB	ATODP
0000H	0150H	PUB	PDECIMAL
0000H	0050H	PUB	X005
0000H	0050H	PUB	M012
0000H	0050H	PUB	M0N4
1000H	0006H	PUB	MAXB
1000H	0051H	PUB	PASS0
1000H	0054H	PUB	PASS1
1000H	005CH	PUB	FC3
1000H	007CH	PUB	CR
1000H	007FH	PUB	R0
1000H	0030H	PUB	TBUFF
1000H	005CH	PUB	FC34
1000H	0102H	PUB	SAVERX
1000H	0106H	PUB	SAVEDX

1000H	0108H	SYM	SYSDATPINTER
1000H	0108H	BAS	SD
1000H	010CH	SYM	CCBPINTER
1000H	0110H	SYM	XINITPINTER
1000H	0110H	BAS	XINIT
0000H	0102H	SYM	PRINTCHAR
0000H	0115H	SYM	PRINTCONSOLEBUFF
			-ER
0000H	0125H	SYM	VERSION
0000H	0143H	SYM	GETCONSOLENUMBER
0000H	0150H	SYM	PDECIMAL
STACK	0006H	SYM	PREC
1000H	0130H	SYM	D
0000H	0215H	SYM	SHOWSTATE
1000H	0114H	SYM	STATE
0000H	028AH	SYM	READCHANGEMXQ
0000H	02BEH	SYM	ATONB
0000H	02F2H	SYM	ATODP
0000H	0312H	SYM	ATOL
1000H	0126H	SYM	ACCUM
1000H	0131H	SYM	VAL
1000H	0133H	SYM	LEN
0000H	03E9H	SYM	COMPARE
STACK	0008H	SYM	OSTR
STACK	0004H	SYM	MAXLEN
1000H	012AH	SYM	CHPLEN
1000H	0134H	SYM	CONSOLE
1000H	0135H	SYM	OPTIONPTR
0000H	0102H	LIN	38
0000H	0111H	LIN	41
0000H	0118H	LIN	44
0000H	0125H	LIN	46
0000H	0134H	LIN	48
0000H	0137H	LIN	50

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P.O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

0000H	0141H	LIN	51
0000H	0146H	LIN	53
0000H	0152H	LIN	55
0000H	015BH	LIN	57
0000H	0160H	LIN	60
0000H	0173H	LIN	62
0000H	0189H	LIN	64
0000H	01A0H	LIN	67
0000H	01ADH	LIN	70
0000H	01B3H	LIN	72
0000H	01C0H	LIN	74
0000H	01D4H	LIN	76
0000H	01E8H	LIN	78
0000H	01FCH	LIN	80
0000H	0210H	LIN	82
0000H	0215H	LIN	84
0000H	0222H	LIN	87
0000H	0235H	LIN	89
0000H	024CH	LIN	91
0000H	0256H	LIN	94
0000H	0272H	LIN	96
0000H	0286H	LIN	99
0000H	028DH	LIN	112
0000H	02A2H	LIN	114
0000H	02A7H	LIN	116
0000H	02BCH	LIN	118
0000H	02C1H	LIN	121
0000H	02D3H	LIN	123
0000H	02E7H	LIN	125
0000H	02F2H	LIN	127
0000H	02F5H	LIN	130
0000H	0307H	LIN	132
0000H	0312H	LIN	134
0000H	0315H	LIN	140
0000H	033DH	LIN	142
0000H	037EH	LIN	144
0000H	0384H	LIN	146
0000H	03C3H	LIN	148
0000H	03CDH	LIN	150
0000H	03D3H	LIN	152
0000H	03E2H	LIN	154
0000H	03E9H	LIN	156
0000H	040DH	LIN	162
0000H	0438H	LIN	164
0000H	044FH	LIN	167
0000H	0458H	LIN	170
0000H	045BH	LIN	177
0000H	046AH	LIN	180
0000H	047AH	LIN	183
0000H	0483H	LIN	185
0000H	04C2H	LIN	187
0000H	04D6H	LIN	189
0000H	04E1H	LIN	192
0000H	0506H	LIN	194
0000H	052BH	LIN	196
0000H	055DH	LIN	200
0000H	0575H	LIN	202
0000H	0589H	LIN	205
0000H	058FH	LIN	208
0000H	0599H	LIN	211

0000H	0143H	LIN	52
0000H	0152H	LIN	54
0000H	0155H	LIN	56
0000H	015DH	LIN	58
0000H	0167H	LIN	61
0000H	017DH	LIN	63
0000H	019BH	LIN	65
0000H	01A4H	LIN	68
0000H	01AFH	LIN	71
0000H	01B6H	LIN	73
0000H	01CAH	LIN	75
0000H	01DEH	LIN	77
0000H	01F2H	LIN	79
0000H	0206H	LIN	81
0000H	0213H	LIN	83
0000H	0218H	LIN	86
0000H	0232H	LIN	88
0000H	023EH	LIN	90
0000H	024EH	LIN	92
0000H	0260H	LIN	95
0000H	027CH	LIN	98
0000H	028AH	LIN	111
0000H	0298H	LIN	113
0000H	02A4H	LIN	115
0000H	02B2H	LIN	117
0000H	02BEH	LIN	119
0000H	02CDH	LIN	122
0000H	02DEH	LIN	124
0000H	02EBH	LIN	126
0000H	02F2H	LIN	128
0000H	0301H	LIN	131
0000H	0308H	LIN	133
0000H	0312H	LIN	135
0000H	0320H	LIN	141
0000H	036CH	LIN	143
0000H	0382H	LIN	145
0000H	038DH	LIN	147
0000H	03CAH	LIN	149
0000H	03D1H	LIN	151
0000H	03DCH	LIN	153
0000H	03E9H	LIN	155
0000H	03ECH	LIN	161
0000H	0434H	LIN	163
0000H	0445H	LIN	166
0000H	0452H	LIN	169
0000H	0458H	LIN	174
0000H	0461H	LIN	178
0000H	0471H	LIN	181
0000H	0494H	LIN	184
0000H	04B6H	LIN	186
0000H	04CFH	LIN	188
0000H	04DCH	LIN	191
0000H	04FAH	LIN	193
0000H	051FH	LIN	195
0000H	0544H	LIN	198
0000H	0569H	LIN	201
0000H	057FH	LIN	204
0000H	058CH	LIN	207
0000H	058FH	LIN	210
0000H	059CH	LIN	213

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P.O. BOX 570

FARMING GROVE, CA 93950

SER. # _____

0000H 05A3H LIN 215
0000H 0102H LIN 217

0000H 05A6H LIN 216

MEMORY MAP OF MODULE SLD
READ FROM FILE VCMODE.LNK
WRITTEN TO FILE VCMODE.

SEGMENT MAP

START	STOP	LENGTH	ALIGN	NAME	CLASS
00000H	00547H	0548H	G	CODE	CODE
10000H	10107H	0108H	G	DATS	DATA
10108H	10135H	002EH	W	DATA	DATA
10136H	10177H	0042H	W	STACK	STACK
10178H	103C3H	024CH	W	CONST	CONST
103D0H	103D0H	0000H	G	??SEG	
103D0H	103D0H	0000H	W	MEMORY	MEMORY

GROUP MAP

ADDRESS	GROUP OR SEGMENT NAME
00000H	CGROUP
	CODE
10000H	DGROUP
	DATS
	STACK
	CONST
	DATA

