

```
1  /* filedump */
2  /*****
3  /*
4  /* Module:   filedump
5  /*         Peter Flass -- Aug 2007
6  /*
7  /* Function: Dump a ENV(VARLS) file.
8  /*
9  /* Usage:   filedump <filename>
10 /*          If <filename> does not exist,
11 /*          UNDEFINEDFILE will be raised.
12 /*
13 /* Dependencies:
14 /*          PL/I record key format.
15 /*
16 /* To Do:   Allow starting and ending record numbers
17 /*          to be specified.
18 /*
19 /*****
20 filedump: proc(parm) options(main);
21   dcl parm          char(*)          varying;
22   dcl sysprint      print;
23   dcl input         input record keyed env(V recsize(32767) );
24   dcl buf           char(32767)      varying;
25   dcl count         fixed bin(31)    init(0);
26   dcl eof           bit(1)          init( '0'b );
27   dcl p             ptr;
28   dcl l             fixed bin(31);
29   dcl char4         char(4);
30
31   dcl VarStr        char(0)          varying based;
32
33   dcl key           char(12);        /* 'record key'      */
34   dcl 1 keyx        based,          /*
35     2 key_addr      fixed bin(31),   /* Record disk address */
36     2 fil          fixed bin(31),   /* Reserved            */
37     2 key_len      fixed bin(31),   /* Record length+4    */
38     2 key_end      char(0);         /* End of record key  */
39
40   on endfile(input) eof='1'b;
41
42   open file(input) title(parm);
43   read file(input) into(buf) keyto(key);
44
45   do while( ^eof );
46     count = count+1;
47     l = length(buf);
48     char4 = bsw( addr(l) );
49     put skip edit( 'Record ', count, ' length ', l,
50     ' ('' || hex(addr(char4)+1,3) || ''x)' )
51     (a,p'zzzz9',a,p'zzzz9',a);
52     p = addr(buf)+stg(null->varstr);
53     call hexdump( p, p+length(buf), addr(key)->key_addr, 0 );
54     read file(input) into(buf) keyto(key);
55     end; /* do while */
56   close file(input);
57   put skip edit( count, ' records read' )(p'zzzz9',a);
58
```

```
59 /*-----*/
60 /*      Hex dump of a specified area      */
61 /*-----*/
62 hexdump: proc(from,to,disk_addr,offset);
63   dcl  (from,to)      ptr;
64   dcl  (disk_addr,offset)  fixed bin(31);
65   dcl  (p,q)          ptr;
66   dcl  (i,n,o,a,x)    fixed bin(31);
67   dcl  line           char(80) varying;
68   dcl  dmask          char(80) static   var  init(
69   /*      .....          .....          */
70   ( 'xxxxxxxx ( xxxxxx)          |'
71   | | (17)' ' ) );
72   dcl  dword          ptr          based;
73   dcl  c4             char(4)     based;
74
75   p = from;           /* Starting memory address */
76   a = disk_addr;     /* Starting disk address */
77   o = offset;        /* Starting offset */
78   q = addr(line) + 56 + /* ->Char output area */
79   stg(null()->VarStr);
80   n = 20;
81   do while( p<to );
82     if n=20 then do; /* Plug address info */
83       line = dmask; /* Initialize dump line */
84       char4 = bsw( addr(a) ); /* Address */
85       substr(line,1,8) = HEX(addr(char4),4);
86       i = o; /* Offset (+|-) */
87       if i<0 then i=-i;
88       char4 = bsw( addr(i) );
89       substr(line,12,6) = HEX(addr(char4)+1,3);
90       if o<0 then substr(line,11,1)='-';
91     end;
92     if to-p<4 then x = to-p;
93     else x=4;
94     substr(line,n,8) = HEX(p,x); /* Data, no byte swapping */
95     q->c4 = xlt(substr(p->c4,1,x)); /* Four 4-byte dwords */
96     n = n+9;
97     q = q+4;
98     if n>54 then do;
99       substr(q->c4,1,1)='|';
100      put skip edit(line)(a);
101      q = addr(line) + 56 +
102      stg(null()->VarStr);
103      substr(q->c4,1,17)='';
104      n = 20;
105      end; /* n */
106      a = a+stg(p->dword);
107      p = p+stg(p->dword);
108      o = o+stg(p->dword);
109      end; /* do while */
110     if n>20 then do;
111       substr(q->c4,1,1)='|';
112       put skip edit(line)(a);
113       end;
114     put skip;
115   end hexdump;
116
```

```
117 /*-----*/
118 /*      Swap byte-order for display      */
119 /*-----*/
120 bsw: proc(p) returns( char(4) );
121     dcl    p          ptr;
122     dcl    cx          (4)char(1) based(p);
123     dcl    cy          (4)char(1);
124     cy(4) = cx(1);      /* OR cy = reverse(cx) */
125     cy(1) = cx(4);
126     cy(3) = cx(2);
127     cy(2) = cx(3);
128     return( string(cy) );
129     end bsw;
130
131 /*-----*/
132 /*      Translate characters to printable  */
133 /*-----*/
134 xlt: proc(c4) returns( char(4) );
135     dcl    c4          char(4);
136     dcl    c4x         char(4);
137     dcl    i          fixed bin(7);
138     c4x = c4;
139     do i=1 to 4;
140         if substr(c4x,i,1) < ' ' |
141            (substr(c4x,i,1) > byte(127) &
142             substr(c4x,i,1) = byte(172))
143         then substr(c4x,i,1) = '.';
144     end;
145     return( c4x );
146     end xlt;
147
148 /*-----*/
149 /*      Hexadecimal to character conversion */
150 /*-----*/
151
152 HEX: proc(pChs,iFB15) returns( char(256) varying );
153     dcl    pChs        ptr;
154     dcl    iFB15       fixed bin(15);
155     dcl    s           char(4096) based(pChs);
156     dcl    j           fixed bin(15);
157     dcl    ret         char(256) varying init('');
158
159     do j=1 to iFB15;
160         ret=ret||hexone( substr(s,j,1) );
161     end;
162     return(ret);
163
164 hexone: proc(c) returns( char(2) );
165     dcl    c           char;
166     dcl    hextabs     char(16) static init
167                ('0123456789ABCDEF'),
168                (0:15)char(1) defined hextabs;
169     dcl    p           ptr,
170            x           bit(8) based(p);
171     p = addr(c);
172     return( hextab( substr(x,1,4) ) || hextab( substr(x,5,4) ) );
173     end hexone;
174
```

Feb 22, 2020 16:14

/* filedump

*/

Page 4

0.9.10a

Source Listing

175 end HEX;

176

177 end filedump;

A	66	76	84	106								
ADDR	48	48	49	52	53	78	84	85	88	89	101	
		171										
BSW	120	48	84	88								
BUF	24	43	47	52	53	54						
BYTE	140	140										
C	165	171										
C4	73	95	99	103	111							
C4	135	138										
C4X	136	138	140	143	145							
CHAR4	29	48	49	84	85	88	89					
COUNT	25	46	49	57								
CX	122	124	125	126	127							
CY	123	124	125	126	127	128						
DISK_ADDR	64	76										
DMASK	68	83										
DWORD	72	106	107	108								
EOF	26	40	45									
FIL	36											
FILEDUMP	20											
FROM	63	75										
HEX	152	49	85	89	94							
HEXDUMP	62	53										
HEXONE	164	160										
HEXTAB	168	172										
HEXTABS	166	168										
I	66	86	87	88								
I	137	139	140	143								
IFB15	154	159										
INPUT	23	40	42	43	54	56						
J	156	159	160									
KEY	33	43	53	54								
KEYX	34											
KEY_ADDR	35	53										
KEY_END	38											
KEY_LEN	37											
L	28	47	48	49								
LENGTH	47	47	53									
LINE	67	78	83	85	89	90	94	100	101	112		
N	66	80	82	94	96	98	104	110				
NULL	52	52	78	101								
O	66	77	86	90	108							
OFFSET	64	77										
P	27	52	53									
P	65	75	81	92	94	95	106	107	108			
P	121	122										
P	169	170	171									
PARM	21	42										
PCHS	153	155										
Q	65	78	95	97	99	101	103	111				
RET	157	160	162									
S	155	160										
STG	52	52	78	101	106	107	108					
STRING	128	128										
SUBSTR	85	85	89	90	94	95	99	103	111	140	143	
		160	172									
SYSPRINT	22	49	57	100	112	114						

Feb 22, 2020 16:14

/* filedump

*/

0.9.10a

Symbol Table and Cross-Reference Listing

Page 6

TO	63	81	92		
VARSTR	31	52	78	101	
X	66	92	93	94	95
X	170	172			
XLT	134	95			