

FFFFFFFFFF	IIIIII	LL	EEEEEEEEEE	SSSSSSSS	
FFFFFFFFFF	IIIIII	LL	EEEEEEEEEE	SSSSSSSS	
FF	II	LL	EE	SS	
FF	II	LL	EE	SS	
FF	II	LL	EE	SS	
FF	II	LL	EE	SS	
FFFFFFFFFF	II	LL	EEEEEEEEEE	SSSSSS	
FFFFFFFFFF	II	LL	EEEEEEEEEE	SSSSSS	
FF	II	LL	EE	SS	
FF	II	LL	EE	SS	
FF	II	LL	EE	SS	
FF	II	LL	EE	SS	
FF	II	LL	EE	SS	
FF	IIIIII	LLLLLLLLLL	EEEEEEEEEE	SSSSSSSS	
FF	IIIIII	LLLLLLLLLL	EEEEEEEEEE	SSSSSSSS	

LL	IIIIII	SSSSSSSS
LL	IIIIII	SSSSSSSS
LL	II	SS
LL	II	SS
LL	II	SS
LL	II	SS
LL	II	SS
LL	II	SS
LL	II	SS
LL	II	SS
LL	II	SS
LL	II	SS
LL	II	SS
LL	II	SS
LLLLLLLLLL	IIIIII	SSSSSSSS
LLLLLLLLLL	IIIIII	SSSSSSSS


```
1 0001 0 MODULE
2 0002 0 FILES (IDENT = 'V04-000') =
3 0003 1 BEGIN
4 0004 1
5 0005 1
6 0006 1 *****
7 0007 1 *
8 0008 1 * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
9 0009 1 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
10 0010 1 * ALL RIGHTS RESERVED.
11 0011 1 *
12 0012 1 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
13 0013 1 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
14 0014 1 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
15 0015 1 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
16 0016 1 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
17 0017 1 * TRANSFERRED.
18 0018 1 *
19 0019 1 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
20 0020 1 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
21 0021 1 * CORPORATION.
22 0022 1 *
23 0023 1 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
24 0024 1 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
25 0025 1 *
26 0026 1 *
27 0027 1 *****
28 0028 1
29 0029 1 ++
30 0030 1 FACILITY: ACC, Account file dumper
31 0031 1
32 0032 1 ABSTRACT:
33 0033 1
34 0034 1 This module contains the file manipulation code for
35 0035 1 the accounting utilities.
36 0036 1
37 0037 1 ENVIRONMENT:
38 0038 1
39 0039 1 VAX/VMS operating system. unprivileged user mode,
40 0040 1
41 0041 1 AUTHOR: Greg Robert and Steve Forgey, January 1982
42 0042 1
43 0043 1 Modified by:
44 0044 1
45 0045 1 V03-003 DAS0003 David Solomon 28-Feb-1984
46 0046 1 Remove related NAM block from INPUT_NAM (in conjunction with
47 0047 1 edit V03-016 to ACC.B32, which causes LIB$FILE_SCAN to do the
48 0048 1 sticky filespec parsing).
49 0049 1
50 0050 1 V03-002 DAS0001 David Solomon 12-Jan-1984
51 0051 1 Get ACCDEF.REQ from SRC$, not MSRC$.
52 0052 1
53 0053 1 V03-001 SPF0105 Steve Forgey 27-Mar-1982
54 0054 1 Set up a related name block for input files.
55 0055 1
56 0056 1 V02-002 SPF0079 Steve Forgey Feb-06-1982
57 0057 1 If /BINARY is not present use ".LIS" as output file default.
```

FILES
V04-000

C 13
15-Sep-1984 23:40:33
14-Sep-1984 11:52:02

VAX-11 Bliss-32 V4.0-742
DISK\$VMSMASTER:[ACC.SRC]FILES.B32;1

Page 2
(1)

```

: 58      0058 1 | otherwise use ".DAT".
: 59      0059 1 |
: 60      0060 1 | V02-001 SPF0071      Steve Forgey      Jan-23-1982
: 61      0061 1 |      Change input file name defaults and improve I/O performance.
: 62      0062 1 |
: 63      0063 1 | --
: 64      0064 1 |
: 65      0065 1 | -----+
: 66      0066 1 |
: 67      0067 1 | INCLUDE FILES
: 68      0068 1 |
: 69      0069 1 | -----+
: 70      0070 1 |
: 71      0071 1 | REQUIRE 'SRC$:ACCDEF';      ! Common ACC definitions
```


73	0910	1	-----
74	0911	1	
75	0912	1	TABLE OF CONTENTS
76	0913	1	-----
77	0914	1	
78	0915	1	
79	0916	1	-----
80	0917	1	
81	0918	1	GENERAL STORAGE DEFINITIONS
82	0919	1	-----
83	0920	1	
84	0921	1	
85	0922	1	
86	0923	1	OWN
87	0924	1	
88	0925	1	DATEXT: INITIAL ('.DAT'),
89	0926	1	! ".DAT" extension
90	0927	1	LISEXT: INITIAL ('.LIS'),
91	0928	1	! ".LIS" extension
92	0929	1	INPUT_NAM_RESULT:
93	0930	1	! Resultant input name
94	0931	1	VECTOR [NAM\$C_MAXRSS,BYTE],
95	0932	1	! -allocate storage
96	0933	1	INPUT_NAM_EXPANDED:
97	0934	1	! Expanded input name
98	0935	1	VECTOR [NAM\$C_MAXRSS,BYTE],
99	0936	1	! -allocate storage
100	0937	1	RELATED_NAM_RESULT:
101	0938	1	! Resultant related name
102	0939	1	VECTOR [NAM\$C_MAXRSS,BYTE],
103	0940	1	! -allocate storage
104	0941	1	OUTPUT_NAM_RESULT:
105	0942	1	! Expanded output name
106	0943	1	VECTOR [NAM\$C_MAXRSS,BYTE],
107	0944	1	! -allocate storage
108	0945	1	REJECTED_NAM_RESULT:
109	0946	1	! Resultant rejected name
110	0947	1	VECTOR [NAM\$C_MAXRSS,BYTE],
111	0948	1	! -allocate storage
112	0949	1	REJECTED_NAM_EXPANDED:
113	0950	1	! Expanded rejected name
114	P 0951	1	GLOBAL
115	P 0952	1	RELATED_NAM: \$NAM(
116	0953	1	Related NAM block
117	0954	1	! -file name address after opening
118	P 0955	1	! -(buffer size)
119	P 0956	1	INPUT_NAM: \$NAM(
120	P 0957	1	Input NAM block
121	P 0958	1	! -file name address after parsing
122	0959	1	! -(buffer size)
123	0960	1	! -file name address after opening
124	P 0961	1	! -(buffer size)
125	P 0962	1	OUTPUT_NAM: \$NAM(
126	P 0963	1	Output NAM block
127	P 0964	1	! -get further defaults from input
128	P 0965	1	! -file name address after parsing
129	0966	1	! -(buffer size)

```

: 130      0967 1
: 131      P 0968 1 REJECTED_NAM: $NAM(
: 132      P 0969 1     RLF = INPUT_NAM,
: 133      P 0970 1     ESA = REJECTED_NAM_EXPANDED,
: 134      P 0971 1     ESS = NAM$C_MAXRSS,
: 135      P 0972 1     RSA = REJECTED_NAM_RESULT,
: 136      0973 1     RSS = NAM$C_MAXRSS,
: 137      0974 1
: 138      0975 1 INPUT_XABFHC: $XABFHC(),
: 139      0976 1
: 140      P 0977 1 INPUT_FAB: $FAB(
: 141      P 0978 1     XAB = INPUT_XABFHC,
: 142      P 0979 1     FOP = (SQO),
: 143      P 0980 1     SHR = (PUT, UPI),
: 144      P 0981 1     NAM = INPUT_NAM,
: 145      P 0982 1     DNM = 'ACCOUNTNG.DAT',
: 146      0983 1     FAC = GET),
: 147      0984 1
: 148      P 0985 1 INPUT_RAB: $RAB(
: 149      P 0986 1     USZ = 512,
: 150      P 0987 1     MBC = 16,
: 151      P 0988 1     MBF = 2,
: 152      P 0989 1     ROP = (RAH),
: 153      P 0990 1     CTX = MSG$_READERR,
: 154      0991 1     FAB = INPUT_FAB),
: 155      0992 1
: 156      P 0993 1 OUTPUT_FAB: $FAB(
: 157      P 0994 1     CTX = MSG$_OPENOUT,
: 158      P 0995 1     FOP = (OFP, SQO),
: 159      P 0996 1     NAM = OUTPUT_NAM,
: 160      P 0997 1     DNS = 4,
: 161      0998 1     DNA = DATEXT),
: 162      0999 1
: 163      P 1000 1 OUTPUT_RAB: $RAB(
: 164      P 1001 1     CTX = MSG$_WRITEERR,
: 165      1002 1     FAB = output_fab),
: 166      1003 1
: 167      P 1004 1 REJECTED_FAB: $FAB(
: 168      P 1005 1     DNM = 'REJ',
: 169      P 1006 1     CTX = MSG$_OPENOUT,
: 170      P 1007 1     FOP = (OFP, SQO),
: 171      1008 1     NAM = REJECTED_NAM),
: 172      1009 1
: 173      P 1010 1 REJECTED_RAB: $RAB(
: 174      P 1011 1     CTX = MSG$_WRITEERR,
: 175      P 1012 1     MBC = 16,
: 176      P 1013 1     MBF = 2,
: 177      P 1014 1     ROP = (WBH),
: 178      1015 1     FAB = REJECTED_FAB);

```

```

: Rejected NAM block
:   -related file name
:   -file name address after parsing
:   -(buffer size)
:   -file name address after open
:   -(buffer size)
:
: Input FHC XAB block
:
: Input FAB block
:   -address of FHC XAB block
:   -sequential operations only
:   -allow un-interlocked, sharing
:   -address of NAM block
:   -default name
:   -open for input
:
: Input RAB block
:   -(buffer size)
:   -multi-block count
:   -multi-buffer count
:   -read-ahead processing
:   -error message value
:   -address of FAB to be CONNECTed
:
: Output FAB block
:   -error message value
:   -output file parse, sequential only
:   -address of NAM block
:   -default extension size
:   -default extension address
:
: Output RAB block
:   -specify error message
:   -address of FAB block
:
: Rejected FAB block
:   -default extension
:   -error message value
:   -output file parse, sequential only
:   -address of NAM block
:
: Rejected RAB block
:   -specify error message
:   -multi-block count
:   -multi-buffer count
:   -write behind processing
:   -address of FAB block

```



```

180 1016 1 UNDECLARE PARSE_OUTPUT_FILES;
181 1017
182 1018 1 GLOBAL ROUTINE PARSE_OUTPUT_FILES =
183 1019 1
184 1020 1 ----
185 1021 1
186 1022 1 Functional description
187 1023 1
188 1024 1 This routine is called to process output files.
189 1025 1 If the files are binary (/BINARY or /REJECTED)
190 1026 1 RMS is used, else the screen package.
191 1027 1
192 1028 1 Input parameters
193 1029 1
194 1030 1 None
195 1031 1
196 1032 1 Output parameters
197 1033 1
198 1034 1 Any errors encountered are RETURNed immediately.
199 1035 1 TRUE is returned on a normal exit.
200 1036 1
201 1037 1 ----
202 1038 1
203 1039 2 BEGIN
204 1040 2
205 1041 2 LOCAL
206 1042 2 desc: vector [2, long]; ! Temporary string descriptor
207 1043 2
208 1044 2 GLOBAL
209 1045 2 screen_char: bblock [scr$k_length]; ! Holds screen characteristics
210 1046 2
211 1047 2 OWN
212 1048 2 output_desc: bblock [dsc$k_d_bln]
213 1049 2 preset([dsc$b_c[lass] = dsc$k_class_d),
214 1050 2 rejected_desc: bblock [dsc$k_d_bln]
215 1051 2 preset([dsc$b_c[lass] = dsc$k_class_d);
216 1052 2
217 1053 2
218 1054 2
219 1055 2 ! PARSE COMMAND LINE OUTPUTS ---
220 1056 2 Parse the /OUTPUT and /REJECTED command switches. Store any output
221 1057 2 file names obtained in the FAB for future processing.
222 1058 2
223 1059 2
224 1060 2 GET_VALUE ('OUTPUT', output_desc);
225 1061 2
226 1062 2 output_fab [fab$b_fns] = .output_desc [dsc$w_length];
227 1063 2 output_fab [fab$l_fna] = .output_desc [dsc$a_pointer];
228 1064 2
229 1065 2 If PRESENT (BINARY)
230 1066 2
231 1067 2 then BEGIN
232 1068 2 perform ($create ( ! Call RMS with
233 1069 2 fab = output_fab, ! -address of FAB
234 1070 2 err = log_filename)); ! -error action routine
235 1071 2
236 1072 2 perform ($connect ( ! Call RMS with

```

```

: 237 P 1073 3 rab = output_rab, ; -address of RAB
: 238 1074 3 err = log_filename)); ; -error action routine
: 239 1075 3 END
: 240 1076 3
: 241 1077 3 else BEGIN
: 242 1078 3 output_fab [fab$l_dna] = lisext; ; Use ".LIS" extension with /NOBINARY
: 243 P 1079 3 perform ($parse ( ; Call RMS parse with
: 244 1080 3 fab = output_fab, ; -address of FAB
: 245 1081 3 err = log_filename)); ; -error action routine
: 246 1082 3 output_desc [dsc$w_length] = .output_nam [nam$b_esl];
: 247 1083 3 output_desc [dsc$a_pointer] = .output_nam [nam$t_esa];
: 248 1084 3 SET OUTPUT (1,output_desc); ; Establish output stream #1
: 249 1085 3 SCREEN_INFO (screen_char); ; Get characteristics
: 250 1086 3 END;
: 251 1087 3
: 252 1088 3
: 253 1089 3
: 254 1090 3 If GET_VALUE ('REJECTED', rejected_desc) then! /REJECTED value
: 255 1091 3 BEGIN
: 256 1092 3 rejected_fab [fab$b_fns] = .rejected_desc [dsc$w_length];
: 257 1093 3 rejected_fab [fab$l_fna] = .rejected_desc [dsc$a_pointer];
: 258 P 1094 3 perform ($create ( ; Call RMS with
: 259 1095 3 fab = rejected_fab, ; -address of FAB
: 260 1096 3 err = log_filename)); ; -error action routine
: 261 1097 3
: 262 P 1098 3 perform ($connect ( ; Call RMS with
: 263 P 1099 3 rab = rejected_rab, ; -address of RAB
: 264 1100 3 err = log_filename)); ; -error action routine
: 265 1101 3 END;
: 266 1102 3
: 267 1103 3 RETURN TRUE;
: 268 1104 3 END;

```

```

.TITLE FILES
.IDENT \V04-000\
.PSECT DATA,NOEXE,2

```

```

54 41 44 2E 00000 DATEXT: .ASCII \.DAT\
53 49 4C 2E 00004 LISEXT: .ASCII \.LIS\
00008 INPUT_NAM RESULT:
          .BLKB 255
00107          .BLKB 1
00108 INPUT_NAM EXPANDED:
          .BLKB 255
00207          .BLKB 1
00208 RELATED_NAM RESULT:
          .BLKB 255
00307          .BLKB 1
00308 OUTPUT_NAM RESULT:
          .BLKB 255
00407          .BLKB 1
00408 OUTPUT_NAM EXPANDED:
          .BLKB 255
00507          .BLKB 1
00508 REJECTED_NAM_RESULT:

```



```

00607 .BLKB 255
00608 REJECTED_NAM_EXPANDED:
00707 .BLKB 255
02 00708 RELATED_NAM::
60 00709 .BYTE 2
FF 0070A .BYTE 96
00 0070B .BYTE -1
00000000 0070C .BYTE 0
00 00710 .ADDRESS RELATED_NAM_RESULT
00 00711 .BYTE 0
00 00712 .BYTE 0
00 00713 .BYTE 0
00000000 00714 .LONG 0
00000000 00718 .LONG 0
0000# 0071C .WORD 0[8]
0000# 0072C .WORD 0[3]
0000# 00732 .WORD 0[3]
00000000 00738 .LONG 0
00000000 0073C .LONG 0
00 00740 .BYTE 0
00 00741 .BYTE 0
00 00742 .BYTE 0
00 00743 .BYTE 0
00 00744 .BYTE 0
00 00745 .BYTE 0
00# 00746 .BYTE 0[2]
00000000 00748 .LONG 0
00000000 0074C .LONG 0
00000000 00750 .LONG 0
00000000 00754 .LONG 0
00000000 00758 .LONG 0
00000000 0075C .LONG 0
00000000# 00760 .LONG 0[2]
02 00768 INPUT_NAM::
60 00769 .BYTE 2
FF 0076A .BYTE 96
00 0076B .BYTE -1
00000000 0076C .BYTE 0
00 00770 .ADDRESS INPUT_NAM_RESULT
00 00771 .BYTE 0
FF 00772 .BYTE -1
00 00773 .BYTE 0
00000000 00774 .ADDRESS INPUT_NAM_EXPANDED
00000000 00778 .LONG 0
0000# 0077C .WORD 0[8]
0000# 0078C .WORD 0[3]
0000# 00792 .WORD 0[3]
00000000 00798 .LONG 0
00000000 0079C .LONG 0
00 007A0 .BYTE 0
00 007A1 .BYTE 0
00 007A2 .BYTE 0
00 007A3 .BYTE 0

```

```

00 007A4 .BYTE 0
00 007A5 .BYTE 0
00# 007A6 .BYTE 0[2]
00000000 007A8 .LONG 0
00000000 007AC .LONG 0
00000000 007B0 .LONG 0
00000000 007B4 .LONG 0
00000000 007B8 .LONG 0
00000000 007BC .LONG 0
00000000# 007C0 .LONG 0[2]
02 007C8 OUTPUT_NAM::
        .BYTE 2
        60 007C9 .BYTE 96
        FF 007CA .BYTE -1
        00 007CB .BYTE 0
00000000' 007CC .ADDRESS OUTPUT_NAM_RESULT
        00 007D0 .BYTE 0
        00 007D1 .BYTE 0
        FF 007D2 .BYTE -1
        00 007D3 .BYTE 0
00000000' 007D4 .ADDRESS OUTPUT_NAM_EXPANDED
00000000' 007D8 .ADDRESS INPUT_NAM
        0000# 007DC .WORD 0[8]
        0000# 007EC .WORD 0[3]
        0000# 007F2 .WORD 0[3]
00000000 007F8 .LONG 0
00000000 007FC .LONG 0
        00 00800 .BYTE 0
        00 00801 .BYTE 0
        00 00802 .BYTE 0
        00 00803 .BYTE 0
        00 00804 .BYTE 0
        00 00805 .BYTE 0
        00# 00806 .BYTE 0[2]
00000000 00808 .LONG 0
00000000 0080C .LONG 0
00000000 00810 .LONG 0
00000000 00814 .LONG 0
00000000 00818 .LONG 0
00000000 0081C .LONG 0
00000000# 00820 .LONG 0[2]
02 00828 REJECTED_NAM::
        .BYTE 2
        60 00829 .BYTE 96
        FF 0082A .BYTE -1
        00 0082B .BYTE 0
00000000' 0082C .ADDRESS REJECTED_NAM_RESULT
        00 00830 .BYTE 0
        00 00831 .BYTE 0
        FF 00832 .BYTE -1
        00 00833 .BYTE 0
00000000' 00834 .ADDRESS REJECTED_NAM_EXPANDED
00000000' 00838 .ADDRESS INPUT_NAM
        0000# 0083C .WORD 0[8]
        0000# 0084C .WORD 0[3]
        0000# 00852 .WORD 0[3]
00000000 00858 .LONG 0

```



```

00000000 0085C .LONG 0
      00 00860 .BYTE 0
      00 00861 .BYTE 0
      00 00862 .BYTE 0
      00 00863 .BYTE 0
      00 00864 .BYTE 0
      00 00865 .BYTE 0
      00# 00866 .BYTE 0[2]
00000000 00868 .LONG 0
00000000 0086C .LONG 0
00000000 00870 .LONG 0
00000000 00874 .LONG 0
00000C00 00878 .LONG 0
00000000 0087C .LONG 0
00000000# 00880 .LONG 0[2]
      1D 00888 INPUT_XABFHC::
      .BYTE 29
      2C 00889 .BYTE 44
      0000 0088A .WORD 0
00000000 0088C .LONG 0
00000000# 00890 .LONG 0[9]
      03 008B4 INPUT_FAB::
      .BYTE 3
      50 008B5 .BYTE 80
      0000 008B6 .WORD 0
00000040 008B8 .LONG 64
00000000 008BC .LONG 0
00000000 008C0 .LONG 0
00000000 008C4 .LONG 0
      0000 008C8 .WORD 0
      02 008CA .BYTE 2
      41 008CB .BYTE 65
00000000 008CC .LONG 0
      00 008D0 .BYTE 0
      00 008D1 .BYTE 0
      00 008D2 .BYTE 0
      02 008D3 .BYTE 2
00000000 008D4 .LONG 0
00000000' 008D8 .ADDRESS INPUT_XABFHC
00000000' 008DC .ADDRESS INPUT_NAM
00000000' 008E0 .LONG 0
00000000' 008E4 .ADDRESS P.AAA
      00 008E8 .BYTE 0
      0D 008E9 .BYTE 13
      0000 008EA .WORD 0
00000000 008EC .LONG 0
      0000 008F0 .WORD 0
      00 008F2 .BYTE 0
      00 008F3 .BYTE 0
00000000 008F4 .LONG 0
00000000 008F8 .LONG 0
      0000 008FC .WORD 0
      00 008FE .BYTE 0
      00 008FF .BYTE 0
00000000 00900 .LONG 0
      01 00904 INPUT_RAB::
      .BYTE 1

```

44	00905	.BYTE	68
0000	00906	.WORD	0
00000200	00908	.LONG	512
00000000	0090C	.LONG	0
00000000	00910	.LONG	0
0000#	00914	.WORD	0[3]
0000	0091A	.WORD	0
009F10B2	0091C	.LONG	10424498
0000	00920	.WORD	0
00	00922	.BYTE	0
00	00923	.BYTE	0
0200	00924	.WORD	512
0000	00926	.WORD	0
00000000	00928	.LONG	0
00000000	0092C	.LONG	0
00000000	00930	.LONG	0
00000000	00934	.LONG	0
00	00938	.BYTE	0
00	00939	.BYTE	0
02	0093A	.BYTE	2
10	0093B	.BYTE	16
00000000	0093C	.LONG	0
00000000	00940	.ADDRESS	INPUT_FAB
00000000	00944	.LONG	0
03	00948	OUTPUT_FAB::	
		.BYTE	3
50	00949	.BYTE	80
0000	0094A	.WORD	0
20000040	0094C	.LONG	536870976
00000000	00950	.LONG	0
00000000	00954	.LONG	0
00000000	00958	.LONG	0
0000	0095C	.WORD	0
02	0095E	.BYTE	2
00	0095F	.BYTE	0
009F10A2	00960	.LONG	10424482
00	00964	.BYTE	0
00	00965	.BYTE	0
00	00966	.BYTE	0
02	00967	.BYTE	2
00000000	00968	.LONG	0
00000000	0096C	.LONG	0
00000000	00970	.ADDRESS	OUTPUT_NAM
00000000	00974	.LONG	0
00000000	00978	.ADDRESS	DATEXT
00	0097C	.BYTE	0
04	0097D	.BYTE	4
0000	0097E	.WORD	0
00000000	00980	.LONG	0
0000	00984	.WORD	0
00	00986	.BYTE	0
00	00987	.BYTE	0
00000000	00988	.LONG	0
00000000	0098C	.LONG	0
0000	00990	.WORD	0
00	00992	.BYTE	0
00	00993	.BYTE	0


```

00000000 00994 .LONG 0
01 00998 OUTPUT_RAB::
      44 00999 .BYTE 1
0000 0099A .BYTE 68
00000000 0099C .WORD 0
00000000 009A0 .LONG 0
00000000 009A4 .LONG 0
0000# 009A8 .WORD 0[3]
0000 009AE .WORD 0
009F10D2 009B0 .LONG 10424530
0000 009B4 .WORD 0
00 009B6 .BYTE 0
00 009B7 .BYTE 0
0000 009B8 .WORD 0
0000 009BA .WORD 0
00000000 009BC .LONG 0
00000000 009C0 .LONG 0
00000000 009C4 .LONG 0
00000000 009C8 .LONG 0
00 009CC .BYTE 0
00 009CD .BYTE 0
00 009CE .BYTE 0
00 009CF .BYTE 0
00000000 009D0 .LONG 0
00000000 009D4 .ADDRESS OUTPUT_FAB
00000000 009D8 .LONG 0
03 009DC REJECTED_FAB::
      50 009DD .BYTE 3
0000 009DE .BYTE 80
20000040 009E0 .WORD 0
00000000 009E4 .LONG 536870976
00000000 009E8 .LONG 0
00000000 009EC .LONG 0
0000 009F0 .WORD 0
02 009F2 .BYTE 2
00 009F3 .BYTE 0
009F10A2 009F4 .LONG 10424482
00 009F8 .BYTE 0
00 009F9 .BYTE 0
00 009FA .BYTE 0
02 009FB .BYTE 2
00000000 009FC .LONG 0
00000000 00A00 .LONG 0
00000000 00A04 .ADDRESS REJECTED_NAM
00000000 00A08 .LONG 0
00000000 00A0C .ADDRESS P.AAB
00 00A10 .BYTE 0
04 00A11 .BYTE 4
0000 00A12 .WORD 0
00000000 00A14 .LONG 0
0000 00A18 .WORD 0
00 00A1A .BYTE 0
00 00A1B .BYTE 0
00000000 00A1C .LONG 0
00000000 00A20 .LONG 0

```

```

0000 00A24 .WORD 0
00 00A26 .BYTE 0
00 00A27 .BYTE 0
00000000 00A28 .LONG 0
01 00A2C REJECTED_RAB::
00A2D .BYTE 1
44 00A2D .BYTE 68
0000 00A2E .WORD 0
00000400 00A30 .LONG 1024
00000000 00A34 .LONG 0
00000000 00A38 .LONG 0
0000# 00A3C .WORD 0[3]
0000 00A42 .WORD 0
009F10D2 00A44 .LONG 10424530
0000 00A48 .WORD 0
00 00A4A .BYTE 0
00 00A4B .BYTE 0
0000 00A4C .WORD 0
0000 00A4E .WORD 0
00000000 00A50 .LONG 0
00000000 00A54 .LONG 0
00000000 00A58 .LONG 0
00000000 00A5C .LONG 0
00 00A60 .BYTE 0
00 00A61 .BYTE 0
02 00A62 .BYTE 2
10 00A63 .BYTE 16
00000000 00A64 .LONG 0
00000000 00A68 .ADDRESS REJECTED_FAB
00000000 00A6C .LONG 0
00A70 SCREEN_CHAR::
00# 00A84 OUTPUT_DESC:
00A87 .BYTE 0[3]
02 00A87 .BYTE 2
00A88 .BLKB 4
00# 00A8C REJECTED_DESC:
00A8F .BYTE 0[3]
02 00A8F .BYTE 2
00A90 .BLKB 4

```

```

54 41 44 2E 47 4E 54 4E 55 4F 43 43 41 00000 P.AAA: .ASCII \ACCOUNTNG.DAT\
4A 45 52 2E 0000D P.AAB: .ASCII \.REJ\
00011 .BLKB 3
00 00 54 55 50 54 55 4F 00014 P.AAD: .ASCII \OUTPUT\<0><0>
00000006 0001C P.AAC: .LONG 6
00000000 00020 .ADDRESS P.AAD
44 45 54 43 45 4A 45 52 00024 P.AAF: .ASCII \REJECTED\
00000008 0002C P.AAE: .LONG 8
00000000 00030 .ADDRESS P.AAF

```

```

.EXTRN ACC$ INVACCREC, ACC$ TOTAL
.EXTRN ACC$ MERGE, ACC$ INPUT
.EXTRN ACC$ TITLETRUNC
.EXTRN ADD_SYMBOL, ALLOCATE

```


.ENTRY	PARSE OUTPUT_FILES, Save R2,R3,R4,R5,R6,R7	: 1018
MOVAB	SYSSCONNECT, R7	:
MOVAB	SYSSCREATE, R6	:
MOVAB	CLISGET_VALUE, R5	:
MOVAB	LOG_FILENAME, R4	:
MOVAB	OUTPUT_DESC, R3	:
SUBL2	#8, SP-	:
PUSHL	R3	: 1060
PUSHAB	P.AAC	:
CALLS	#2, CLISGET_VALUE	:
MOVB	OUTPUT_DESC, OUTPUT_FAB+52	: 1062
MOVL	OUTPUT_DESC+4, OUTPUT_FAB+44	: 1063
BBC	#3, QUALIFIERS, 1\$	: 1065
PUSHL	R4	: 1070
PUSHAB	OUTPUT_FAB	:
CALLS	#2, SYSSCREATE	:
BLBC	STATUS, 2\$	:
PUSHL	R4	: 1074
PUSHAB	OUTPUT_RAB	:
CALLS	#2, SYSSCONNECT	:
BLBS	STATUS, 4\$	:
RET		:
MOVAB	LISEXT, OUTPUT_FAB+48	: 1078
PUSHL	R4	: 1081
PUSHAB	OUTPUT_FAB	:
CALLS	#2, SYSSPARSE	:
BLBC	STATUS, 6\$	:
MOVZBW	OUTPUT_NAM+11, OUTPUT_DESC	: 1082
MOVL	OUTPUT_NAM+12, OUTPUT_DESC+4	: 1083

			7E	7C	0007E	CLRQ	-(SP)		1084
			7E	D4	00080	CLRL	-(SP)		
			53	DD	00082	PUSHL	R3		
00000000G	00		01	DD	00084	PUSHL	#1		
	52		05	FB	00086	CALLS	#5, SCR\$SET_OUTPUT		
	10		50	D0	0008D	MOVL	R0, STATUS		
			52	E9	00090	BLBC	STATUS, 3\$		
		EC	A3	9F	00093	PUSHAB	SCREEN_CHAR		1085
00000000G	00		01	FB	00096	CALLS	#1, SCR\$SCREEN_INFO		
	52		50	D0	0009D	MOVL	R0, STATUS		
	0D		52	E8	000A0	BLBS	STATUS, 4\$		
			52	DD	000A3	PUSHL	STATUS		
00000000G	00		01	FB	000A5	CALLS	#1, LIB\$SIGNAL		
	50		52	D0	000AC	MOVL	STATUS, R0		
				04	000AF	RET			
		08	A3	9F	000B0	PUSHAB	REJECTED_DESC		1090
		FF41	CF	9F	000B3	PUSHAB	P.AAE		
	65		02	FB	000B7	CALLS	#2, CLISGET_VALUE		
	21		50	E9	000BA	BLBC	R0, 5\$		
8C	A3	08	A3	90	000BD	MOVB	REJECTED_DESC, REJECTED_FAB+52		1092
84	A3	0C	A3	D0	000C2	MOVL	REJECTED_DESC+4, REJECTED_FAB+44		1093
			54	DD	000C7	PUSHL	R4		1096
		FF58	C3	9F	000C9	PUSHAB	REJECTED_FAB		
	66		02	FB	000CD	CALLS	#2, SYSS\$CREATE		
	0E		50	E9	000D0	BLBC	STATUS, 6\$		
			54	DD	000D3	PUSHL	R4		1100
		A8	A3	9F	000D5	PUSHAB	REJECTED_RAB		
	67		02	FB	000D8	CALLS	#2, SYSS\$CONNECT		
	03		50	E9	000DB	BLBC	STATUS, 6\$		
	50		01	D0	000DE	MOVL	#1, R0		1103
				04	000E1	RET			1104

; Routine Size: 226 bytes, Routine Base: CODE + 0034

FILES
V04-000

C 14
15-Sep-1984 23:40:33
14-Sep-1984 11:52:02

VAX-11 Bliss-32 V4.0-742
DISK\$VMSMASTER:[ACC.SRC]FILES.B32;1

Page 15
(4)

: 270 1105 1 END
: 271 1106 0 ELUDOM

.EXTRN LIB\$SIGNAL

PSECT SUMMARY

Name	Bytes	Attributes
DATA	2708	NOVEC, WRT, RD, NOEXE, NOSHR, LCL, REL, CON, NOPIC, ALIGN(2)
CODE	278	NOVEC, NOWRT, RD, EXE, NOSHR, LCL, REL, CON, NOPIC, ALIGN(2)

Library Statistics

File	----- Total	Symbols Loaded	----- Percent	Pages Mapped	Processing Time
_\$255\$DUA28:[SYSLIB]STARLET.L32;1	9776	76	0	581	00:01.0
_\$255\$DUA28:[SYSLIB]TPAMAC.L32;1	42	0	0	14	00:00.1

COMMAND QUALIFIERS

: BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/LIS=LIS\$:FILES/OBJ=OBJ\$:FILES MSRC\$:FILES/UPDATE=(ENH\$:FILES)

: Size: 226 code + 2760 data bytes
: Run Time: 00:18.1
: Elapsed Time: 00:54.2
: Lines/CPU Min: 3670
: Lexemes/CPU-Min: 46098
: Memory Used: 186 pages
: Compilation Complete

0001 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY