

FFFFFFFFFFFFFFFF	111	111	XXX	XXX
FFFFFFFFFFFFFFFF	111	111	XXX	XXX
FFFFFFFFFFFFFFFF	111	111	XXX	XXX
FFF	111111	111111	XXX	XXX
FFF	111111	111111	XXX	XXX
FFF	111111	111111	XXX	XXX
FFF	111	111		
FFF	111	111	XXX	XXX
FFF	111	111	XXX	XXX
FFFFFFFFF.FFF	111	111		
FFFFFFFFFFFFFFFF	111	111	XXX	
FFFFFFFFFFFFFFFF	111	111	XXX	
FFF	111	111		
FFF	111	111	XXX	XXX
FFF	111	111	XXX	XXX
FFF	111	111	XXX	XXX
FFF	111	111		
FFF	111	111	XXX	XXX
FFF	111	111	XXX	XXX
FFF	1111111111	1111111111	XXX	XXX
FFF	1111111111	1111111111	XXX	XXX
FFF	1111111111	1111111111	XXX	XXX

.....

```
PPPPPPPP      AAAAAA      RRRRRRRR      SSSSSSSS      NN      NN      MM      MM
PPPPPPPP      AAAAAA      RRRRRRRR      SSSSSSSS      NN      NN      MM      MM
PP      PP      AA      AA      RR      RR      SS      NN      NN      MMMM      MMMM
PP      PP      AA      AA      RR      RR      SS      NN      NN      MMMM      MMMM
PP      PP      AA      AA      RR      RR      SS      NNNN      NN      MM      MM
PPPPPPPP      AA      AA      RRRRRRRR      SSSSSS      NN      NN      NN      MM      MM
PPPPPPPP      AA      AA      RRRRRRRR      SSSSSS      NN      NN      NN      MM      MM
PP      AAAAAAAAAA      RR      RR      SS      NN      NNNN      MM      MM
PP      AAAAAAAAAA      RR      RR      SS      NN      NNNN      MM      MM
PP      AA      AA      RR      RR      SS      NN      NN      MM      MM
PP      AA      AA      RR      RR      SSSSSSSS      NN      NN      MM      MM
PP      AA      AA      RR      RR      SSSSSSSS      NN      NN      MM      MM
```

```
....
....
....
....
```

```
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
LLLLLLLLLLLL      IIIIII      SSSSSSSS
LLLLLLLLLLLL      IIIIII      SSSSSSSS
```

```
1 0001 0 MODULE PARSNM (  
2 0002 0 LANGUAGE (BLISS32),  
3 0003 0 IDENT = 'V04-000',  
4 0004 0 ) =  
5 0005 1 BEGIN  
6 0006 1  
7 0007 1  
8 0008 1 *****  
9 0009 1 *  
10 0010 1 * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY  
11 0011 1 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.  
12 0012 1 * ALL RIGHTS RESERVED.  
13 0013 1 *  
14 0014 1 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED  
15 0015 1 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE  
16 0016 1 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER  
17 0017 1 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY  
18 0018 1 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY  
19 0019 1 * TRANSFERRED.  
20 0020 1 *  
21 0021 1 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE  
22 0022 1 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT  
23 0023 1 * CORPORATION.  
24 0024 1 *  
25 0025 1 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS  
26 0026 1 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.  
27 0027 1 *  
28 0028 1 *  
29 0029 1 *****  
30 0030 1  
31 0031 1 ++  
32 0032 1  
33 0033 1 FACILITY: F11ACP Structure Level 2  
34 0034 1  
35 0035 1 ABSTRACT:  
36 0036 1  
37 0037 1 This routine parses a file name string.  
38 0038 1  
39 0039 1 ENVIRONMENT:  
40 0040 1  
41 0041 1 STARLET operating system, including privileged system services  
42 0042 1 and internal exec routines.  
43 0043 1  
44 0044 1 --  
45 0045 1  
46 0046 1  
47 0047 1 AUTHOR: Andrew C. Goldstein, CREATION DATE: 3-Jan-1978 18:30  
48 0048 1  
49 0049 1 MODIFIED BY:  
50 0050 1  
51 0051 1 V03-003 CDS0002 Christian D. Saether 2-Jan-1984  
52 0052 1 Force longword addressing on FIL$CVT_DTB.  
53 0053 1  
54 0054 1 V03-002 CDS0001 Christian D. Saether 6-Dec-1983  
55 0055 1 Change LIB$ references to FIL$.  
56 0056 1  
57 0057 1 V03-001 ACG0302 Andrew C. Goldstein, 3-Dec-1982 11:19
```

PARSNM
V04-000

H 14
16-Sep-1984 00:48:57
14-Sep-1984 12:30:40

VAX-11 Bliss-32 V4.0-742
DISK\$VMSMASTER:[F11X.SRC]PARSNM.B32;1 Page 2 (1)

```

: 58      0058 1 |      Add support for long file names and $ and _
: 59      0059 1 |
: 60      0060 1 |      V02-002 ACG0208      Andrew C. Goldstein,      31-Oct-1981  0:01
: 61      0061 1 |      Allow negative version numbers; interface change for
: 62      0062 1 |      string buffer
: 63      0063 1 |
: 64      0064 1 |      V02-001 ACG0167      Andrew C. Goldstein,      16-Apr-1980  19:27
: 65      0065 1 |      Previous revision history moved to F11B.REV
: 66      0066 1 |      **
: 67      0067 1 |
: 68      0068 1 |
: 69      0069 1 |      LIBRARY 'SYS$LIBRARY:LIB.L32';
: 70      0070 1 |      REQUIRE 'SRC$:FCPDEF.B32';

```

```
1061 1 GLOBAL ROUTINE PARSE_NAME (NAME_DESC, NAME_BUFFER, COUNT, STRING, FLAGS) : NOVALUE =
1062 1
1063 1 ++
1064 1
1065 1 FUNCTIONAL DESCRIPTION:
1066 1
1067 1     This routine parses a file name string.
1068 1
1069 1
1070 1 CALLING SEQUENCE:
1071 1     PARSE_NAME (ARG1, ARG2, ARG3, ARG4, ARG5)
1072 1
1073 1 INPUT PARAMETERS:
1074 1     ARG3: character count of name string
1075 1     ARG4: address of name string
1076 1     ARG5: name control flag bits
1077 1
1078 1 IMPLICIT INPUTS:
1079 1     NONE
1080 1
1081 1 OUTPUT PARAMETERS:
1082 1     ARG1: address of name descriptor block
1083 1     ARG2: address of name string buffer to use
1084 1
1085 1 IMPLICIT OUTPUTS:
1086 1     NONE
1087 1
1088 1 ROUTINE VALUE:
1089 1     NONE
1090 1
1091 1 SIDE EFFECTS:
1092 1     NONE
1093 1
1094 1 --
1095 1
1096 2 BEGIN
1097 2
1098 2 MAP
1099 2     NAME_DESC      : REF BBLOCK,      ! name descriptor block arg
1100 2     NAME_BUFFER    : REF VECTOR [,BYTE]; ! name string buffer arg
1101 2
1102 2 LOCAL
1103 2     DP              : REF BBLOCK,      ! local pointer to descriptor
1104 2     WILD_BIT,        ! copy of wild name and type bit
1105 2     J,               ! character counter
1106 2     P1,              ! pointer to scan string
1107 2     P2,              ! pointer to build output string
1108 2     C                : BYTE,          ! character in process
1109 2     CCOUNT,          ! number of significant chars in string
1110 2     DOT_COUNT,       ! count of "." encountered
1111 2     NEG,             ! flag indicating negative version
1112 2
1113 2 EXTERNAL ROUTINE
1114 2     FIL$CVT_DTB      : ADDRESSING_MODE (GENERAL); ! convert decimal to binary
1115 2
1116 2
1117 2 ! Load a local pointer to the name descriptor block. Once the compiler
```

```
1118 2 ! learns how to address structures through formals, this code can be removed.
1119 2 !
1120 2
1121 2 DP = .NAME_DESC;
1122 2
1123 2 ! Iterate on the characters, copying alphanumerics in upper case.
1124 2 ! If a wild card is present if name or type, we leave that field null.
1125 2 ! Dots and semicolons delimit the name and type strings; once two have been
1126 2 ! seen, get the version number. Note that a semicolon seen without a dot is
1127 2 ! processed twice to produce a null type field. We stuff dots at the end
1128 2 ! if necessary to get the descriptors completed.
1129 2 !
1130 2
1131 2 CH$FILL (0, FND_LENGTH, .DP);
1132 2 DP[FND_FLAGS] = .FLAGS AND NOT $FIELDMASK (FIB$V_WILD);
1133 2 WILD_BIT = .DP[FND_WILD_NAME];
1134 2 J = .COUNT;
1135 2 P1 = .STRING;
1136 2 P2 = .NAME_BUFFER;
1137 2 CCOUNT = 0;
1138 2 DOT_COUNT = 0;
1139 2 DP[FND_STRING] = .P2;
1140 2
1141 2 WHILE 1 DO
1142 2 BEGIN
1143 2 IF (J = .J - 1) GEQ 0
1144 2 THEN
1145 2 C = CH$RCHAR_A (P1)
1146 2 ELSE
1147 2 BEGIN
1148 2 C = '.';
1149 2 J = .J + 1;
1150 2 END;
1151 2
1152 2 IF .P2 = .NAME_BUFFER GTRU FILENAME_LENGTH
1153 2 THEN ERR_EXIT (SS$_BADFILENAME);
1154 2
1155 2 SELECTONEU .C OF
1156 2 SET
1157 2 ['A' TO 'Z', '0' TO '9', '$', '_']:
1158 2 BEGIN
1159 2 CCOUNT = .CCOUNT + 1;
1160 2 IF NOT .WILD_BIT
1161 2 THEN CH$WCHAR_A (.C, P2);
1162 2 END;
1163 2
1164 2 ['a' TO 'z']:
1165 2 BEGIN
1166 2 C = .C AND NOT %X'20';
1167 2 CCOUNT = .CCOUNT + 1;
1168 2 IF NOT .WILD_BIT
1169 2 THEN CH$WCHAR_A (.C, P2);
1170 2 END;
1171 2
1172 2 ['.', ';']:
1173 2 BEGIN
1174 2
```

```
186 1175 4 IF .CCOUNT GTRU 39
187 1176 4 THEN ERR_EXIT (SS$_BADFILENAME);
188 1177 4 IF NOT .DOT_COUNT
189 1178 4 THEN
190 1179 5 BEGIN
191 1180 5 IF .WILD_BIT
192 1181 5 THEN
193 1182 6 BEGIN
194 1183 6 DP[FND_WILD] = 1;
195 1184 6 CH$WCHAR_A ('*', P2);
196 1185 5 END;
197 1186 5 IF .C EQL '; '
198 1187 5 THEN
199 1188 6 BEGIN
200 1189 6 P1 = .P1 - 1;
201 1190 6 J = .J + 1;
202 1191 5 END;
203 1192 5 CH$WCHAR_A ('.', P2);
204 1193 5 WILD_BIT = .DP[FND_WILD_TYPE];
205 1194 5 CCOUNT = 0;
206 1195 5 DOT_COUNT = .DOT_COUNT + 1;
207 1196 5 END
208 1197 5
209 1198 4 ELSE
210 1199 5 BEGIN
211 1200 5 IF .WILD_BIT
212 1201 5 THEN
213 1202 6 BEGIN
214 1203 6 DP[FND_WILD] = 1;
215 1204 6 CH$WCHAR_A ('*', P2);
216 1205 5 END;
217 1206 5 EXITLOOP;
218 1207 4 END;
219 1208 3
220 1209 3
221 1210 4 ['X']: BEGIN
222 1211 4 DP[FND_WILD] = 1;
223 1212 4 CCOUNT = .CCOUNT + 1;
224 1213 4 IF NOT .WILD_BIT
225 1214 4 THEN CH$WCHAR_A (.C, P2);
226 1215 3 END;
227 1216 3
228 1217 4 ['*']: BEGIN
229 1218 4 DP[FND_WILD] = 1;
230 1219 4 IF CH$RCHAR (.P1-2) NEQ '*'
231 1220 4 AND NOT .WILD_BIT
232 1221 4 THEN CH$WCHAR_A (.C, P2);
233 1222 3 END;
234 1223 3
235 1224 3
236 1225 3 [OTHERWISE]:
237 1226 3 ERR_EXIT (SS$_BADFILENAME);
238 1227 3
239 1228 3 TES;
240 1229 3
241 1230 3 END;
242 1231 2 ! end of character loop
```

```
243 1232 2 ! Record the length of the generated string. Then convert and check the
244 1233 2 ! version number.
245 1234 2 !
246 1235 2 !
247 1236 2 DP[FND_COUNT] = .P2 - .DP[FND_STRING];
248 1237 2 IF .J EQL 1
249 1238 2 AND CH$RCHAR (.P1) EQL '*'
250 1239 2 THEN DP[FND_WILD_VER] = 1;
251 1240 2
252 1241 2 NEG = 0;
253 1242 2 IF NOT .DP[FND_WILD_VER]
254 1243 2 THEN
255 1244 2 BEGIN
256 1245 2 IF .J GTR 0
257 1246 2 AND CH$RCHAR (.P1) EQL '-'
258 1247 2 THEN
259 1248 2 BEGIN
260 1249 2 NEG = 1;
261 1250 2 J = .J - 1;
262 1251 2 P1 = .P1 + 1;
263 1252 2 END;
264 1253 2 IF NOT FIL$CVT_DTB (.J, .P1, DP[FND_VERSION])
265 1254 2 THEN ERR_EXIT (SS$_BADFILEVER);
266 1255 2 END;
267 1256 2
268 1257 2 IF .(DP[FND_VERSION]) < 0, 32 > GTRU 32768
269 1258 2 THEN ERR_EXIT (SS$_BADFILEVER);
270 1259 2
271 1260 2 IF .NEG
272 1261 2 THEN
273 1262 2 BEGIN
274 1263 2 DP[FND_VERSION] = -.DP[FND_VERSION];
275 1264 2 IF .DP[FND_VERSION] EQL 0
276 1265 2 THEN DP[FND_VERSION] = -32768;
277 1266 2 END;
278 1267 2
279 1268 1 END;
```

! end of routine PARSE_NAME

					01FC 00000
10	00	56	04	AC	D0 00002
		6E		00	2C 00006
				66	0000B
	66	14	AC	0100	8F AB 0000C
54	66		01	05	EF 00013
			57	0C	AC D0 00018
			55	10	AC D0 0001C
			50	08	AC D0 00020
				53	D4 00024
				58	D4 00026

.TITLE	PARSNM	
.IDENT	\V04-000\	
.EXTRN	FIL\$CVT_DTB	
.PSECT	\$CODE\$,NOWRT,2	
.ENTRY	PARSE_NAME, Save R2,R3,R4,R5,R6,R7,R8	: 1061
MOVL	NAME_DESC, DP	: 1121
MOVCS	#0, (SP), #0, #16, (DP)	: 1131
BICW3	#256, FLAGS, (DP)	: 1132
EXTZV	#5, #1, (DP), WILD_BIT	: 1133
MOVL	COUNT, J	: 1134
MOVL	STRING, P1	: 1135
MOVL	NAME_BUFFER, P2	: 1136
CLRL	CCOUNT	: 1137
CLRL	DOT_COUNT	: 1138

08	A6	50	D0	00028	MOVL	P2, 8(DP)	1139
		57	D7	0002C	DECL	J	1143
		05	19	0002E	BLSS	2\$	
	51	85	90	00030	MOVB	(P1)+, C	1145
		05	11	00033	BRB	3\$	
	51	2E	90	00035	MOVB	#46, C	1148
		57	D6	00038	INCL	J	1149
52	00000050	50	AC	C3	0003A	3\$:	1152
		8F	52	D1	0003F	SUBL3	NAME BUFFER, P2, R2
			3F	1A	00046	CMPL	R2, #80
		24	51	91	00048	BGTRU	8\$
			75	13	0004B	CMPB	C, #36
		30	51	91	0004D	BEQL	13\$
			05	1F	00050	CMPB	C, #48
		39	51	91	00052	BLSSU	4\$
			6B	1B	00055	CMPB	C, #57
41	8F	51	91	00057	BLEQU	13\$	
		06	1F	0005B	CMPB	C, #65	
5A	8F	51	91	0005D	BLSSU	5\$	
		5F	1B	00061	CMPB	C, #90	
5F	8F	51	91	00063	BLEQU	13\$	
		59	13	00067	CMPB	C, #95	
61	8F	51	91	00069	BEQL	13\$	
		0B	1F	0006D	CMPB	C, #97	1165
7A	8F	51	91	0006F	BLSSU	6\$	
		05	1A	00073	CMPB	C, #122	
		51	20	8A	00075	BGTRU	6\$
		48	11	00078	BICB2	#32, C	1167
		2E	51	91	0007A	BRB	13\$
			05	13	0007D	CMPB	C, #46
		3B	51	91	0007F	BEQL	7\$
			35	12	00082	CMPB	C, #59
		27	53	D1	00084	BNEQ	12\$
			55	1A	00087	CMPL	CCOUNT, #39
		21	58	E8	00089	BGTRU	17\$
		07	54	E9	0008C	BLBS	DOT_COUNT, 11\$
01	A6	01	88	0008F	BLBC	WILD_BIT, 9\$	1177
	80	2A	90	00093	BISB2	#1, T(DP)	1180
	3B	51	91	00096	MOVB	#42, (P2)+	1183
		04	12	00099	CMPB	C, #59	1184
		55	D7	0009B	BNEQ	10\$	1186
		57	D6	0009D	DECL	P1	
		2E	90	0009F	INCL	J	1189
54	66	80	04	EF	000A2	MOVB	#46, (P2)+
		01	53	D4	000A7	EXTZV	#4, #1, (DP), WILD_BIT
			58	D6	000A9	CLRL	CCOUNT
			2E	11	000AB	INCL	DOT_COUNT
		33	54	E9	000AD	BRB	16\$
			01	88	000B0	BLBC	WILD_BIT, 18\$
01	A6	2A	90	000B4	BISB2	#1, T(DP)	1200
	80	2A	11	000B7	MOVB	#42, (P2)+	1203
		25	51	91	000B9	BRB	18\$
			08	12	000BC	CMPB	C, #37
			01	88	000BE	BNEQ	14\$
01	A6	53	D6	000C2	BISB2	#1, 1(DP)	1210
		0F	11	000C4	INCL	CCOUNT	1211
		2A	51	91	000C6	BRB	15\$
					CMPB	C, #42	1212
							1213
							1217

01	A6	13	12	000C9	BNEQ	17\$	
	2A	01	88	000CB	BISB2	#1, 1(DP)	1218
		FE	A5	91	CMPB	-2(P1), #42	1219
	03	06	13	000D3	BEQL	16\$	
	80	54	E8	000D5	BLBS	WILD BIT, 16\$	1220
		51	90	000D8	MOVB	C (P2)+	1221
		FF4E	31	000DB	BRW	1\$	1155
		0818	8F	000DE	CHMU	#2072	1226
			04	000E2	RET		
04	A6	50	08	A6	C3	000E3	18\$: 1236
		01	57	D1	000E9		1237
			08	12	000EC		
	2A	65	91	000EE	BNEQ	19\$	
		03	12	000F1	CMPB	(P1), #42	1238
	66	08	88	000F3	BNEQ	19\$	
		53	D4	000F6	BISB2	#8, (DP)	1239
21	66	03	E0	000F8	CLRL	NEG	1241
		57	D5	000FC	BBS	#3, (DP), 21\$	1242
		0C	15	000FE	TSTL	J	1245
	2D	65	91	00100	BLEQ	20\$	
		07	12	00103	CMPB	(P1), #45	1246
	53	01	D0	00105	BNEQ	20\$	
		57	D7	00108	MOVL	#1, NEG	1249
		55	D6	0010A	DECL	J	1250
		0C	A6	9F	0010C	INCL	P1
			55	DD	0010F	PUSHAB	12(DP)
			57	DD	00111	PUSHL	P1
00000000G	00	03	FB	00113	PUSHL	J	
00008000	0A	50	E9	0011A	CALLS	#3, FIL\$CVT_DTB	
	8F	0C	A6	D1	0011D	BLBC	R0, 22\$
			05	1B	00125	CMPB	12(DP), #32768
		0820	8F	BF	00127	BLEQU	23\$
			04	0012B	CHMU	#2080	1258
	0D	53	E9	0012C	RET		
0C	A6	0C	A6	AE	0012F	BLBC	NEG, 24\$
			06	12	00134	MNEGW	12(DP), 12(DP)
0C	A6	8000	8F	B0	00136	BNEQ	24\$
			04	0013C	24\$: MOVW	#-32768, 12(DP)	1265
					RET		1268

; Routine Size: 317 bytes, Routine Base: \$CODE\$ + 0000

: 280 1269 1
: 281 1270 1 END
: 282 1271 0 ELUDOM

PSECT SUMMARY

Name	Bytes	Attributes
\$CODE\$	317	NOVEC,NOWRT, RD , EXE,NOSHR, LCL, REL, CON,NOPI,ALIGN(2)

PARSNM
V04-000

B 15
16-Sep-1984 00:48:57
14-Sep-1984 12:30:40

VAX-11 BLISS-32 V4.0-742
DISK\$VMSMASTER:[F11X.SRC]PARSNM.B32;1 Page 9 (2)

Library Statistics

File	----- Total	Symbols Loaded	----- Percent	Pages Mapped	Processing Time
;\$255\$DUA28:[SYSLIB]LIB.L32;1	18619	20	0	1000	00:01.9

COMMAND QUALIFIERS

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

; Size: 317 code + 0 data bytes
; Run Time: 00:13.8
; Elapsed Time: 00:34.4
; Lines/CPU Min: 5522
; Lexemes/CPU-Min: 23539
; Memory Used: 216 pages
; Compilation Complete

0171 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

