

```

PPPPPPPPPPPP      AAAAAAAAAA      SSSSSSSSSSSS      RRRRRRRRRRRR      TTTTTTTTTTTTTTTT      LLL
PPPPPPPPPPPP      AAAAAAAAAA      SSSSSSSSSSSS      RRRRRRRRRRRR      TTTTTTTTTTTTTTTT      LLL
PPPPPPPPPPPP      AAAAAAAAAA      SSSSSSSSSSSS      RRRRRRRRRRRR      TTTTTTTTTTTTTTTT      LLL
PPP                PPP  AAA          AAA  SSS                RRR          RRR          TTT          LLL
PPP                PPP  AAA          AAA  SSS                RRR          RRR          TTT          LLL
PPP                PPP  AAA          AAA  SSS                RRR          RRR          TTT          LLL
PPP                PPP  AAA          AAA  SSS                RRR          RRR          TTT          LLL
PPP                PPP  AAA          AAA  SSS                RRR          RRR          TTT          LLL
PPP                PPP  AAA          AAA  SSS                RRR          RRR          TTT          LLL
PPPPPPPPPPPP      AAA          AAA          SSSSSSSSSS      RRRRRRRRRRRR      TTT          LLL
PPPPPPPPPPPP      AAA          AAA          SSSSSSSSSS      RRRRRRRRRRRR      TTT          LLL
PPPPPPPPPPPP      AAA          AAA          SSSSSSSSSS      RRRRRRRRRRRR      TTT          LLL
PPP                AAAAAAAAAAAAAAAAAA  SSS          RRR  RRR          TTT          LLL
PPP                AAAAAAAAAAAAAAAAAA  SSS          RRR  RRR          TTT          LLL
PPP                AAAAAAAAAAAAAAAAAA  SSS          RRR  RRR          TTT          LLL
PPP                AAA          AAA          SSS          RRR  RRR          TTT          LLL
PPP                AAA          AAA          SSS          RRR  RRR          TTT          LLL
PPP                AAA          AAA          SSS          RRR  RRR          TTT          LLL
PPP                AAA          AAA          SSS          RRR  RRR          TTT          LLL
PPP                AAA          AAA          SSSSSSSSSS      RRR          RRR          TTT          LLLLLLLLLLLLLLLLLL
PPP                AAA          AAA          SSSSSSSSSS      RRR          RRR          TTT          LLLLLLLLLLLLLLLLLL
PPP                AAA          AAA          SSSSSSSSSS      RRR          RRR          TTT          LLLLLLLLLLLLLLLLLL

```



```

1 0001 0 MODULE PASSWRITE_VARYING ( %TITLE 'Write a varying string'
2 0002 0 IDENT = '1-002' ! File: PASWRIVAR.B32 Edit: SBL1002
3 0003 0 ) =
4 0004 1 BEGIN
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 **
31 0031 1 FACILITY: Pascal Language Support
32 0032 1
33 0033 1 ABSTRACT:
34 0034 1
35 0035 1 This module contains a procedure which writes a varying string
36 0036 1 to a textfile.
37 0037 1
38 0038 1 ENVIRONMENT: User mode - AST reentrant
39 0039 1
40 0040 1 AUTHOR: Steven B. Lionel, CREATION DATE: 1-April-1981
41 0041 1
42 0042 1 MODIFIED BY:
43 0043 1
44 0044 1 1-001 - Original. SBL 1-April-1981
45 0045 1 1-002 - Make total-width a longword. SBL 30-June-1982
46 0046 1 --
47 0047 1

```

PASSWRITE_VARYI Write a varying string
1-002 Declarations

I 13
16-Sep-1984 02:27:40
14-Sep-1984 12:52:11

VAX-11 Bliss-32 V4.0-742
[PASRTL.SRC]PASWRIVAR.B32;1

Page 2
(2)

```

: 49      0048 1 %SBTTL 'Declarations'
: 50      0049 1
: 51      0050 1 : PROLOGUE DEFINITIONS:
: 52      0051 1 :
: 53      0052 1
: 54      0053 1 REQUIRE 'RTLIN:PASPROLOG';           ! Externals, linkages, PSECTs, structures
: 55      0117 1
: 56      0118 1 :
: 57      0119 1 : TABLE OF CONTENTS:
: 58      0120 1 :
: 59      0121 1
: 60      0122 1 FORWARD ROUTINE
: 61      0123 1     PASSWRITE_VARYING: NOVALUE,       ! Write to textfile
: 62      0124 1     PASSWRITEV_VARYING: NOVALUE;     ! Write to string
: 63      0125 1
: 64      0126 1 :
: 65      0127 1 : MACROS:
: 66      0128 1
: 67      0129 1     NONE
: 68      0130 1
: 69      0131 1 : EQUATED SYMBOLS:
: 70      0132 1
: 71      0133 1     NONE
: 72      0134 1
: 73      0135 1 : FIELDS:
: 74      0136 1
: 75      0137 1     NONE
: 76      0138 1
: 77      0139 1 : OWN STORAGE:
: 78      0140 1
: 79      0141 1     NONE
: 80      0142 1 :
```

PASSWRITE_VARYI
1-002

Write a varying string

PASSWRITE_VARYING - Write a varying string to t

```

82      0143 1 %SBTTL 'PASSWRITE VARYING - Write a varying string to textfile'
83      0144 1 GLOBAL ROUTINE PASSWRITE VARYING (
84      0145 1     PFV: REF $PASSPFV FILE VARIABLE,           ! File variable
85      0146 1     STRING: REF VECTOR [,WORD],              ! Address of string
86      0147 1     ERROR,                                    ! Error unwind address
87      0148 1     TOTAL_WIDTH: SIGNED                      ! Total field width
88      0149 1 ): NOVALUE =
89      0150 1
90      0151 1 ++
91      0152 1 FUNCTIONAL DESCRIPTION:
92      0153 1     This procedure writes a varying string to the specified textfile.
93      0154 1
94      0155 1 CALLING SEQUENCE:
95      0156 1     CALL PASSWRITE_VARYING (PFV.mr.r, STRING.rvt.r
96      0157 1     [, [ERROR.j.r] [, TOTAL_WIDTH.rl.v]])
97      0158 1
98      0159 1 FORMAL PARAMETERS:
99      0160 1
100     0161 1     PFV - The Pascal File Variable (PFV) passed by reference.
101     0162 1     The structure of the PFV is defined in PASPFV.REQ.
102     0163 1
103     0164 1     STRING - The address of the string to write. The string's
104     0165 1     current length is in the first word of the string.
105     0166 1
106     0167 1     ERROR - Optional. If specified, the address to unwind to
107     0168 1     in case of an error.
108     0169 1
109     0170 1     TOTAL_WIDTH - Optional. Total field width, defaults to string length.
110     0171 1
111     0172 1 IMPLICIT INPUTS:
112     0173 1
113     0174 1     NONE
114     0175 1
115     0176 1 IMPLICIT OUTPUTS:
116     0177 1
117     0178 1     NONE
118     0179 1
119     0180 1 ROUTINE VALUE:
120     0181 1
121     0182 1     NONE
122     0183 1
123     0184 1 SIDE EFFECTS:
124     0185 1
125     0186 1     If the file is the standard file INPUT or OUTPUT, it is implicitly opened.
126     0187 1
127     0188 1 SIGNALLED ERRORS:
128     0189 1
129     0190 1     LINTOOLON - line too long
130     0191 1     NEGWIDDIG - negative Width or Digits specification is not allowed
131     0192 1
132     0193 1 --
133     0194 1
134     0195 1 BEGIN
135     0196 1
136     0197 2
137     0198 2
138     0199 2 LOCAL

```

PASSWRITE_VARYI
1-002

Write a varying string
PASSWRITE_VARYING - Write a varying string to t

K 13
16-Sep-1984 02:27:40
14-Sep-1984 12:52:11

VAX-11 Bliss-32 V4.0-742
[PASRTL.SRC]PASWRIVAR.B32;1

Page 4
(3)

```
139      0200      2          FCB: REF $PASSFCB_CONTROL_BLOCK, ! File Control block
140      0201      2          FIELD_WIDTH, ! Total width of field
141      0202      2          PFV_ADDR: VOLATILE, ! Enable argument
142      0203      2          UNWIND_ACT: VOLATILE, ! Enable argument
143      0204      2          ERROR_ADDR: VOLATILE; ! Enable argument
144      0205
145      0206      2          BUILTIN
146      0207      2          ACTUALCOUNT; ! Count of arguments
147      0208
148      0209      2          ENABLE
149      0210      2          PASS$IO_HANDLER (PFV_ADDR, UNWIND_ACT, ERROR_ADDR); ! Enable error handler
150      0211
151      0212      2          !+
152      0213      2          ! Get ERROR parameter, if present.
153      0214      2          !-
154      0215
155      0216      2          IF ACTUALCOUNT () GEQU 3
156      0217      2          THEN
157      0218      2          ERROR_ADDR = .ERROR; ! Set unwind address
158      0219
159      0220      2          PFV_ADDR = PFV [PFV$R_PFV]; ! Set PFV address
160      0221
161      0222      2          !+
162      0223      2          ! Validate PFV and get FCB.
163      0224      2          !-
164      0225
165      0226      2          PASS$VALIDATE_PFV (PFV [PFV$R_PFV]; FCB);
166      0227
167      0228      2          !+
168      0229      2          ! Set unwind action to unlock file.
169      0230      2          !-
170      0231
171      0232      2          UNWIND_ACT = PASS$K_UNWIND_UNLOCK;
172      0233
173      0234      2          !+
174      0235      2          ! Do common initialization.
175      0236      2          !-
176      0237
177      0238      2          PASS$INIT_WRITE (PFV [PFV$R_PFV], FCB [FCB$R_FCB]; FCB);
178      0239
179      0240      2          !+
180      0241      2          ! See if we have enough room in the record.
181      0242      2          !-
182      0243
183      0244      2          IF ACTUALCOUNT () GEQU 4
184      0245      2          THEN
185      0246      2          BEGIN
186      0247      2          FIELD_WIDTH = .TOTAL_WIDTH;
187      0248      2          IF .FIELD_WIDTH LSS 0
188      0249      2          THEN
189      0250      2          $PASSIO_ERROR (PASS_NEGWIDDIG,0);
190      0251      2          END
191      0252      2          ELSE
192      0253      2          FIELD_WIDTH = .STRING [0]; ! Length is in first word
193      0254
194      0255      2          BEGIN
195      0256      2          LOCAL
```


PASSWRITE_VARYI Write a varying string
1-002

PASSWRITE_VARYING - Write a varying string to t

M 13
16-Sep-1984 02:27:40
14-Sep-1984 12:52:11

VAX-11 Bliss-32 V4.0-742
[PASRTL.SRC]PASWRIVAR.B32;1

Page 6
(3)

		6D	0083	CF	DE	00011		MOVAL	6\$, (FP)	
		03		6C	91	00016		CMPB	(AP), #3	0216
				04	1F	00019		BLSSU	1\$	
		6E	0C	AC	D0	0001B		MOVL	ERROR, ERROR_ADDR	0218
		56	04	AC	D0	0001F	1\$:	MOVL	PFV, R6	0220
08		AE		56	D0	00023		MOVL	R6, PFV_ADDR	
			00000000G	00	16	00027		JSB	PASS\$VALIDATE_PFV	0226
04		AE		01	D0	0002D		MOVL	#1, UNWIND_ACT	0232
			00000000G	00	16	00031		JSB	PASS\$INIT_WRITE	0238
		04		6C	91	00037		CMPB	(AP), #4	0244
				10	1F	0003A		BLSSU	2\$	
		58	10	AC	D0	0003C		MOVL	TOTAL_WIDTH, FIELD_WIDTH	0247
				0E	18	00040		BGEQ	3\$	0248
				7E	D4	00042		CLRL	-(SP)	0250
		7E	00G	8F	9A	00044		MOVZBL	#PASSK_NEGWIDDIG, -(SP)	
		6A		02	FB	00048		CALLS	#2, PASS\$SIGNAL	
					04	0004B		RET		
		58	08	BC	3C	0004C	2\$:	MOVZWL	@STRING, FIELD_WIDTH	0253
	50	58	EC	A7	C1	00050	3\$:	ADDL3	-20(FCB), FIELD_WIDTH, R0	0258
		50	FO	A7	C2	00055		SUBL2	-16(FCB), EXTRA	
				0C	15	00059		BLEQ	4\$	0259
				5C	DD	0005B		PUSHL	EXTRA	0261
				01	DD	0005D		PUSHL	#1	
		7E	00G	8F	9A	0005F		MOVZBL	#PASSK_LINTOOLON, -(SP)	
		6A		03	FB	00063		CALLS	#3, PASS\$SIGNAL	
					04	00066		RET		
		59	08	AC	D0	00067	4\$:	MOVL	STRING, R9	0268
58	69	10		00	ED	0006B		CMPZV	#0, #16, (R9), FIELD_WIDTH	
				15	18	00070		BGEQ	5\$	
		50		69	3C	00072		MOVZWL	(R9), R0	0271
	50	58		50	C3	00075		SUBL3	R0, FIELD_WIDTH, R0	
50	20	6E		00	2C	00079		MOVCS	#0, (SP), #32, R0, @-20(FCB)	0272
			EC	B7		0007E				
		EC		A7	D0	00080		MOVL	R3, -20(FCB)	
		58		69	3C	00084		MOVZWL	(R9), FIELD_WIDTH	0273
	EC	02	B7	58	28	00087	5\$:	MOVCS	FIELD_WIDTH, 2(R9), @-20(FCB)	0280
		EC		A7	D0	0008D		MOVL	R3, -20(FCB)	
					16	00091		JSB	PASS\$END_WRITE	0287
					04	00097		RET		0291
					0000	00098	6\$:	.WORD	Save nothing	0197
		50	08	AC	D0	0009A		MOVL	8(AP), R0	
		50	04	A0	D0	0009E		MOVL	4(R0), R0	
			F4	A0	9F	000A2		PUSHAB	ERROR_ADDR	
			F8	A0	9F	000A5		PUSHAB	UNWIND_ACT	
			FC	A0	9F	000AB		PUSHAB	PFV_ADDR	
				03	DD	000AB		PUSHL	#3	
				5E	DD	000AD		PUSHL	SP	
		7E	04	AC	7D	000AF		MOVQ	4(AP), -(SP)	
		00000000G	00	03	FB	000B3		CALLS	#3, PASS\$IO_HANDLER	
					04	000BA		RET		

; Routine Size: 187 bytes, Routine Base: _PASSCODE + 0000

; 231 0292 1
; 232 0293 1 !<BLF/PAGE>


```

234 0294 1 %SBTTL 'PASSWRITEV VARYING - Write varying to string'
235 0295 1 GLOBAL ROUTINE PASSWRITEV_VARYING (
236 0296 1     MAX_LENGTH: WORD,           ! Maximum length of string
237 0297 1     STRING_LINE: REF VECTOR [, WORD], ! String to write to
238 0298 1     STRING: REF VECTOR [, WORD],      ! String to write
239 0299 1     ERROR,                  ! Error unwind address
240 0300 1     TOTAL_WIDTH: SIGNED      ! Total field width
241 0301 1 ) : NOVALUE =
242 0302 1
243 0303 1
244 0304 1 +-+
245 0305 1 FUNCTIONAL DESCRIPTION:
246 0306 1     This procedure writes a varying to the specified string.
247 0307 1
248 0308 1 CALLING SEQUENCE:
249 0309 1
250 0310 1     CALL PASSWRITEV_STRING (MAX_LENGTH.rw.v, STRING_LINE.wvt.r,
251 0311 1     STRING.rvt.r [, [ERROR.j.r] [TOTAL_WIDTH.rl.v]])
252 0312 1
253 0313 1 FORMAL PARAMETERS:
254 0314 1
255 0315 1     MAX_LENGTH - The maximum length of STRING_LINE.
256 0316 1
257 0317 1     STRING_LINE - A varying string to which the output will be appended.
258 0318 1
259 0319 1     STRING - The varying string to write.
260 0320 1
261 0321 1     ERROR - Optional. If specified, the address to unwind to
262 0322 1     in case of an error.
263 0323 1
264 0324 1     TOTAL_WIDTH - The width of the field to write. Optional, defaults
265 0325 1     to STRING_LENGTH.
266 0326 1
267 0327 1 IMPLICIT INPUTS:
268 0328 1
269 0329 1     NONE
270 0330 1
271 0331 1 IMPLICIT OUTPUTS:
272 0332 1
273 0333 1     NONE
274 0334 1
275 0335 1 ROUTINE VALUE:
276 0336 1
277 0337 1     NONE
278 0338 1
279 0339 1 SIDE EFFECTS:
280 0340 1
281 0341 1     NONE
282 0342 1
283 0343 1 SIGNALLED ERRORS:
284 0344 1
285 0345 1     See PASSWRITE_VARYING
286 0346 1
287 0347 1 --
288 0348 1
289 0349 2 BEGIN
290 0350 2

```

```

291 0351 2 LOCAL
292 0352 2 PFV: $PASSPFV FILE VARIABLE, ! Pascal File Variable
293 0353 2 ARG_LIST: VECTOR [5, LONG], ! Argument list
294 0354 2 PFV_ADDR: VOLATILE, ! Enable argument
295 0355 2 UNWIND_ACT: VOLATILE, ! Enable argument
296 0356 2 ERROR_ADDR: VOLATILE; ! Enable argument
297 0357 2
298 0358 2 BUILTIN
299 0359 2 ACTUALCOUNT; ! Count of arguments
300 0360 2
301 0361 2 ENAB E
302 0362 2 PASS$IO_HANDLER (PFV_ADDR, UNWIND_ACT, ERROR_ADDR); ! Enable error handler
303 0363 2
304 0364 2 !+
305 0365 2 ! Get ERROR parameter, if present.
306 0366 2 !-
307 0367 2
308 0368 2 IF ACTUALCOUNT () GEQU 4
309 0369 2 THEN
310 0370 2 ERROR_ADDR = .ERROR; ! Set unwind address
311 0371 2
312 0372 2 PFV_ADDR = PFV [PFV$R_PFV]; ! Set PFV address
313 0373 2
314 0374 2 !+
315 0375 2 ! Set up ARG_LIST.
316 0376 2 !-
317 0377 2
318 0378 2 ARG_LIST [0] = 2; ! Two arguments
319 0379 2 ARG_LIST [1] = PFV [PFV$R_PFV]; ! PFV address
320 0380 2 ARG_LIST [2] = STRING [0]; ! String to write
321 0381 2 IF ACTUALCOUNT () GEQU 5
322 0382 2 THEN
323 0383 2 BEGIN
324 0384 2 ARG_LIST [0] = 4; ! Add two more arguments
325 0385 2 ARG_LIST [3] = 0; ! Error address
326 0386 2 ARG_LIST [4] = .TOTAL_WIDTH; ! Field width
327 0387 2 END;
328 0388 2
329 0389 2 !+
330 0390 2 ! Call PASS$DO_WRITEV to do the work, giving it the address of
331 0391 2 ! PASSWRITE_VARYING to call.
332 0392 2 !-
333 0393 2
334 0394 2 PASS$DO_WRITEV (PFV [PFV$R_PFV], .MAX_LENGTH, STRING_LINE [0], ARG_LIST,
335 0395 2 PASSWRITE_VARYING);
336 0396 2
337 0397 2 RETURN;
338 0398 2
339 0399 1 END; ! End of routine PASSWRITEV_VARYING

```

.EXTRN PASS\$DO_WRITEV

SE 007C 0000
 2C C2 00002
 7E D4 00005

.ENTRY PASSWRITEV_VARYING, Save R2,R3,R4,R5,R6 : 0295
 SUBL2 #44, SP :
 CLRL ERROR_ADDR : 0349

PASSWRITE_VARYI
1-002

Write a varying string
PASSWRITEV_VARYING - Write varying to string

C 14
16-Sep-1984 02:27:40
14-Sep-1984 12:52:11

VAX-11 Bliss-32 V4.0-742
[PASRTL.SRC]PASWRIVAR.B32;1

Page 9
(4)

		04	AE	7C	00007	CLRQ	UNWIND_ACT	
	6D	004A	CF	DE	0000A	MOVAL	3\$, (FP)	
	04		6C	91	0000F	CMPB	(AP), #4	0368
			04	1F	00012	BLSSU	1\$	
	6E	10	AC	D0	00014	MOVL	ERROR, ERROR_ADDR	0370
08	AE	20	AE	9E	00018	MOVAB	PFV, PFV_ADDR	0372
0C	AE		02	D0	0001D	MOVL	#2, ARG_LIST	0378
10	AE	20	AE	9E	00021	MOVAB	PFV, ARG_LIST+4	0379
14	AE	0C	AC	D0	00026	MOVL	STRING, ARG_LIST+8	0380
	05		6C	91	0002B	CMPB	(AP), #5	0381
			0C	1F	0002E	BLSSU	2\$	
0C	AE		04	D0	00030	MOVL	#4, ARG_LIST	0384
		18	AE	D4	00034	CLRL	ARG_LIST+12	0385
1C	AE	14	AC	D0	00037	MOVL	TOTAL WIDTH, ARG_LIST+16	0386
	55	FF05	CF	9E	0003C	MOVAB	PASSWRITE_VARYING, R5	0394
	54	0C	AE	9E	00041	MOVAB	ARG_LIST, -R4	
	56	20	AE	9E	00045	MOVAB	PFV, R6	
	53	08	AC	D0	00049	MOVL	STRING LINE, R3	
	52	04	AC	3C	0004D	MOVZWL	MAX_LENGTH, R2	
		00000000G	00	16	00051	JSB	PASS\$DO_WRITEV	
				04	00057	RET		0399
				0000	00058	.WORD	Save nothing	0349
	50	08	AC	D0	0005A	MOVL	8(AP), R0	
	50	04	A0	D0	0005E	MOVL	4(R0), R0	
		D0	A0	9F	00062	PUSHAB	ERROR_ADDR	
		D4	A0	9F	00065	PUSHAB	UNWIND_ACT	
		D8	A0	9F	00068	PUSHAB	PFV_ADDR	
			03	DD	0006B	PUSHL	#3	
			5E	DD	0006D	PUSHL	SP	
	7E	04	AC	7D	0006F	MOVQ	4(AP), -(SP)	
	00000000G	00	03	FB	00073	CALLS	#3, PASS\$IO_HANDLER	
				04	0007A	RET		

: Routine Size: 123 bytes, Routine Base: _PASS\$CODE + 00BB

: 340 0400 1
: 341 0401 1 !<BLF/PAGE>

PASSWRITE_VARYI Write a varying string
1-002 PASSWRITEV_VARYING - Write varying to string

D 14
16-Sep-1984 02:27:40 VAX-11 Bliss-32 V4.0-742
14-Sep-1984 12:52:11 [PASRTL.SRC]PASWRIVAR.B32;1

: 343 0402 1 END ! End of module PASSWRITE_VARYING
: 344 0403 1
: 345 0404 0 ELUDOM

PSECT SUMMARY

Name Bytes Attributes
:_PASS\$CODE 310 NOVEC,NOWRT, RD , EXE, SHR, LCL, REL, CON, PIC,ALIGN(2)

Library Statistics

File	----- Total	Symbols Loaded	----- Percent	Pages Mapped	Processing Time
_\$255\$DUA28:[SYSLIB]STARLET.L32;1	9776	0	0	581	00:01.0
_\$255\$DUA28:[PASRTL.OBJ]PASLIB.L32;1	427	96	22	33	00:00.4

COMMAND QUALIFIERS

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

: Size: 310 code + 0 data bytes
: Run Time: 00:08.0
: Elapsed Time: 00:16.0
: Lines/CPU Min: 3045
: Lexemes/CPU-Min: 14306
: Memory Used: 91 pages
: Compilation Complete

