


```

LL          IIIIII  BBBB8888  GGGGGGGG  EEEEEEEEEEE  TTTTTTTTTTT  JJ  PPPPPPPP  IIIIII
LL          IIIIII  88888888  GGGGGGGG  EEEEEEEEEEE  TTTTTTTTTTT  JJ  PPPPPPPP  IIIIII
LL          II      88      88  GG          EE          TT
LL          II      88      88  GG          EE          TT
LL          II      88      88  GG          EE          TT
LL          II      88888888  GG          EEEEEEEEE  TT
LL          II      88888888  GG          EEEEEEEEE  TT
LL          II      88      88  GG  GGGGGG  EE          TT
LL          II      88      88  GG  GGGGGG  EE          TT
LL          II      88      88  GG          GG  EE          TT
LL          II      88      88  GG          GG  EE          TT
LL          II      88888888  GGGGGG  EEEEEEEEEEE  TT
LL          IIIIII  88888888  GGGGGG  EEEEEEEEEEE  TT
LL          IIIIII  88888888  GGGGGG

```

```

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          IIIIII  SSSSSSSS
LL          IIIIII  SSSSSSSS

```



```

1 0001 0 MODULE LIB$GETJPI ( %TITLE 'Get Job/Process Information'
2 0002 0 IDENT = '1-003' ! File: LIBGETJPI.B32 Edit: SBL1003
3 0003 0 ) =
4 0004 1 BEGIN
5 0005 1
6 0006 1 *****
7 0007 1 *
8 0008 1 *  COP'RIGHT (c) 1978, 1980, 1982, 1984 BY
9 0009 1 *  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
10 0010 1 *  ALL RIGHTS RESERVED.
11 0011 1 *
12 0012 1 *  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
13 0013 1 *  ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
14 0014 1 *  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
15 0015 1 *  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
16 0016 1 *  OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
17 0017 1 *  TRANSFERRED.
18 0018 1 *
19 0019 1 *  THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
20 0020 1 *  AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
21 0021 1 *  CORPORATION.
22 0022 1 *
23 0023 1 *  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
24 0024 1 *  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
25 0025 1 *
26 0026 1 *
27 0027 1 *****
28 0028 1
29 0029 1
30 0030 1 **
31 0031 1 FACILITY:      General Utility Library
32 0032 1
33 0033 1 ABSTRACT:
34 0034 1
35 0035 1 LIB$GETJPI obtains a specified item of Job/Process information
36 0036 1 and formats it in an appropriate manner.
37 0037 1
38 0038 1 ENVIRONMENT: User mode - AST reentrant
39 0039 1
40 0040 1 AUTHOR: Steven B. Lionel, CREATION DATE: 11-Jan-1983
41 0041 1
42 0042 1 MODIFIED BY:
43 0043 1
44 0044 1 1-001 - Original. SBL 11-Jan-1983
45 0045 1 1-002 - Change format code names to LIB$K_FMT. SBL 11-Mar-1983
46 0046 1 1-003 - Change string length to 512. SBL 11-Mar-1983
47 0047 1 --
48 0048 1

```

```

: 50 0049 1 %SBTTL 'Declarations'
: 51 0050 1
: 52 0051 1 | PROLOGUE FILE:
: 53 0052 1 |
: 54 0053 1 |
: 55 0054 1 REQUIRE 'RTLIN:LIBPROLOG';           ! Switches, PSECTS, macros
: 56 0125 1
: 57 0126 1 |
: 58 0127 1 | LINKAGES:
: 59 0128 1 |
: 60 0129 1 |     NONE
: 61 0130 1 |
: 62 0131 1 | TABLE OF CONTENTS:
: 63 0132 1 |
: 64 0133 1 |
: 65 0134 1 FORWARD ROUTINE
: 66 0135 1 LIB$GETJPI;                          ! Get Job/Process Information
: 67 0136 1
: 68 0137 1 |
: 69 0138 1 | MACROS:
: 70 0139 1 |
: 71 0140 1 |     NONE
: 72 0141 1 |
: 73 0142 1 | EQUATED SYMBOLS:
: 74 0143 1 |
: 75 0144 1 |     NONE
: 76 0145 1 |
: 77 0146 1 | FIELDS:
: 78 0147 1 |
: 79 0148 1 |     NONE
: 80 0149 1 |
: 81 0150 1 | OWN STORAGE:
: 82 0151 1 |
: 83 0152 1 |     NONE
: 84 0153 1 |
: 85 0154 1 | EXTERNALS:
: 86 0155 1 |
: 87 0156 1 |
: 88 0157 1 EXTERNAL ROUTINE
: 89 0158 1 LIB$$GETJPI,                          ! Internal routine
: 90 0159 1 LIB$ANALYZE_SDESC_R2: LIB$ANALYZE_SDESC_R2$LINKAGE, ! Get length and pointer
: 91 0160 1 LIB$GET_EF,                          ! Allocate event flag number
: 92 0161 1 LIB$FREE_EF: NOVALUE,                ! Free event flag number
: 93 0162 1 LIB$SCOPY_R_DX6: LIB$SCOPY_R_DX6$LINKAGE; ! Copy string by reference.
: 94 0163 1
: 95 0164 1 EXTERNAL LITERAL
: 96 0165 1 LIB$_INVARG,                          ! Invalid argument
: 97 0166 1 LIB$_STRTRU;                          ! String truncated

```

```
.. 99 0167 1 %SBTTL 'LIB$GETJPI - Get Job/Process Information'
.. 100 0168 1 GLOBAL ROUTINE LIB$GETJPI (
.. 101 0169 1     ITEM_CODE: REF VECTOR [, WORD],           ! Code of desired item
.. 102 0170 1     PROCESS_ID: REF VECTOR [, LONG],       ! Process identification
.. 103 0171 1     PROCESS_NAME: REF BLOCK [, BYTE],      ! Process name descriptor
.. 104 0172 1     OUT_VALUE: REF VECTOR [, LONG],        ! Output numeric value
.. 105 0173 1     OUT_STRING: REF BLOCK [, BYTE],        ! Output string descriptor
.. 106 0174 1     OUT_LEN: REF VECTOR [, WORD]           ! Output string length
.. 107 0175 1 ) =
.. 108 0176 1
.. 109 0177 1 ++
.. 110 0178 1 | FUNCTIONAL DESCRIPTION:
.. 111 0179 1 |
.. 112 0180 1 | LIB$GETJPI provides a simplified interface to the $GETJPI system
.. 113 0181 1 | service. It provides accounting, status and identification
.. 114 0182 1 | information about a specified process.
.. 115 0183 1 |
.. 116 0184 1 | LIB$GETJPI provides the following features in addition to those
.. 117 0185 1 | provided by the $GETJPI system service:
.. 118 0186 1 |
.. 119 0187 1 |     Instead of a list of item descriptors, which may be
.. 120 0188 1 |     difficult to construct in high-level languages, the single
.. 121 0189 1 |     item desired is specified as an integer code which is passed
.. 122 0190 1 |     by reference. Results are written to separate arguments.
.. 123 0191 1 |
.. 124 0192 1 |     For items which return numeric values, LIB$GETJPI can
.. 125 0193 1 |     optionally provide a formatted string interpretation of the
.. 126 0194 1 |     value. For example, if the process UIC is requested,
.. 127 0195 1 |     LIB$GETJPI can return the UIC formatted as "[g,m]".
.. 128 0196 1 |
.. 129 0197 1 |     For string arguments, all string classes supported by the
.. 130 0198 1 |     Run-Time Library are understood.
.. 131 0199 1 |
.. 132 0200 1 |     Calls to LIB$GETJPI are synchronous. LIB$GETJPI calls
.. 133 0201 1 |     LIB$GET_EF to allocate a local event flag number for
.. 134 0202 1 |     synchronization.
.. 135 0203 1 |
.. 136 0204 1 |     LIB$GETJPI does not provide the ability to obtain more than one
.. 137 0205 1 |     item of information in a single call. See the description of the
.. 138 0206 1 |     $GETJPI system service for more information.
.. 139 0207 1 |
.. 140 0208 1 | CALLING SEQUENCE:
.. 141 0209 1 |
.. 142 0210 1 |     ret-status.wlc.v = LIB$GETJPI (
.. 143 0211 1 |         item-code.rw.r,
.. 144 0212 1 |         [process-id.mlu.r],
.. 145 0213 1 |         [process-name.rt.dx],
.. 146 0214 1 |         [out-value.wz.r]
.. 147 0215 1 |         [, [out-string.wt.dx]
.. 148 0216 1 |         [, [out-len.ww.r] ]])
.. 149 0217 1 |
.. 150 0218 1 | FORMAL PARAMETERS:
.. 151 0219 1 |
.. 152 0220 1 |     item-code
.. 153 0221 1 |     A longword integer item identifier code that specifies which item
.. 154 0222 1 |     of information you are requesting. All valid $GETJPI item codes,
.. 155 0223 1 |     whose names begin with JPI$, are accepted.
```

```
156 0224 1 |
157 0225 1 | process-id
158 0226 1 |   A longword specifying the process identification of the
159 0227 1 |   process for which information is to be returned.  If not
160 0228 1 |   specified, process-name is used.  The longword is updated to
161 0229 1 |   contain the process identification actually used, which may
162 0230 1 |   be different than what was originally specified if
163 0231 1 |   process-name is specified or if "wild card process
164 0232 1 |   searching" is used.
165 0233 1 |
166 0234 1 | process-name
167 0235 1 |   A string specifying the name of the process for which
168 0236 1 |   information is to be returned.  If not specified, process-id
169 0237 1 |   is used.  If neither process-name or process-id are
170 0238 1 |   specified, the caller's process is used.  See the
171 0239 1 |   description of the process-id argument for more information.
172 0240 1 |
173 0241 1 | out-value
174 0242 1 |   A longword or quadword into which is placed the numeric
175 0243 1 |   value of the information requested.  If an item only returns
176 0244 1 |   a string value, this parameter is ignored.
177 0245 1 |
178 0246 1 | out-string
179 0247 1 |   A string into which is placed the string representation of
180 0248 1 |   the information requested.  If out-string is not specified,
181 0249 1 |   and the value returned has only a string representation, the
182 0250 1 |   error status LIB$INVARG is returned.
183 0251 1 |
184 0252 1 | out-len
185 0253 1 |   A word integer into which is placed the number of significant
186 0254 1 |   characters written to out-string, not including blank padding
187 0255 1 |   or truncated characters.
188 0256 1 |
189 0257 1 | IMPLICIT INPUTS:
190 0258 1 |
191 0259 1 |     NONE
192 0260 1 |
193 0261 1 | IMPLICIT OUTPUTS:
194 0262 1 |
195 0263 1 |     NONE
196 0264 1 |
197 0265 1 | COMPLETION STATUS:
198 0266 1 |
199 0267 1 |     $$$ NORMAL      Normal successful completion
200 0268 1 |     LIB$STRTRU      String truncated.  This is an alternate success status.
201 0269 1 |     LIB$INSEF       Insufficient event flags
202 0270 1 |     LIB$INVSTRDES   Invalid string descriptor
203 0271 1 |     LIB$WRONUMARG   Wrong number of arguments
204 0272 1 |     LIB$_xxx        Any error status from LIB$SCOPY_R_DX
205 0273 1 |     $$$BADPARAM     The item code is not recognized as valid.
206 0274 1 |     $$$_xxx        Any error status from $GETJPI
207 0275 1 |
208 0276 1 | SIDE EFFECTS:
209 0277 1 |
210 0278 1 |     NONE
211 0279 1 |
212 0280 1 | --
```

```
214 0281 2 BEGIN
215 0282 2
216 0283 2 LOCAL
217 0284 2 RET_STRING: VECTOR [512, BYTE], ! Local string for value
218 0285 2 RET_LENGTH: WORD, ! Length of RET_STRING
219 0286 2 RET_NUMBER: VECTOR [2, LONG], ! Local quadword for value
220 0287 2 RET_TYPE, ! Returned type code
221 0288 2 LCL_PRCNAM_DSC: BLOCK [8, BYTE], ! Local descriptor for PRCNAM
222 0289 2 PRCNAM_DSC_ADR, ! Address of PRCNAM descriptor
223 0290 2 EVENT_FLAG, ! Event flag number
224 0291 2 STR_STATUS, EF_STATUS, JPI_STATUS, COPY_STATUS; ! Return statuses
225 0292 2
226 0293 2 BUILTIN
227 0294 2 NULLPARAMETER;
228 0295 2
229 0296 2 !+
230 0297 2 ! Validate argument count.
231 0298 2 !-
232 0299 2
233 0300 2 $LIB$VALIDATE_ARGCOUNT (4,6);
234 0301 2
235 0302 2 !+
236 0303 2 ! Build static descriptor for PROCESS_NAME, if any.
237 0304 2 !-
238 0305 2
239 0306 2 IF NULLPARAMETER (3)
240 0307 2 THEN
241 0308 2 PRCNAM_DSC_ADR = 0 ! Omitted
242 0309 2 ELSE
243 0310 2 BEGIN
244 0311 2 LCL_PRCNAM_DSC [DSC$B_DTYPE] = DSC$K_DTYPE_T;
245 0312 2 LCL_PRCNAM_DSC [DSC$B_CLASS] = DSC$K_CLASS_S;
246 0313 2 STR_STATUS = LIB$ANALYZE_SDESC_R2 (PROCESS_NAME [0,0,0,0]);
247 0314 2 LCL_PRCNAM_DSC [DSC$B_LENGTH], LCL_PRCNAM_DSC [DSC$A_POINTER]);
248 0315 2 IF NOT .STR_STATUS
249 0316 2 THEN
250 0317 2 RETURN .STR_STATUS;
251 0318 2 PRCNAM_DSC_ADR = LCL_PRCNAM_DSC;
252 0319 2 END;
253 0320 2
254 0321 2 !+
255 0322 2 ! Allocate an event flag number to use for the $GETJPI.
256 0323 2 !-
257 0324 2
258 0325 2 EF_STATUS = LIB$GET_EF (EVENT_FLAG);
259 0326 2 IF NOT .EF_STATUS
260 0327 2 THEN
261 0328 2 RETURN .EF_STATUS;
262 0329 2
263 0330 2 !+
264 0331 2 ! Call LIB$$GETJPI to do the work.
265 0332 2 !-
266 0333 2
267 0334 2 JPI_STATUS = LIB$$GETJPI (
268 0335 2 .ITEM_CODE [0],
269 0336 2 RET_STRING,
270 0337 2 RET_NUMBER,
```

```
271 0338 2 RET_LENGTH,  
272 0339 2 RET_TYPE,  
273 0340 2 .EVENT_FLAG,  
274 0341 2 PROCESS_ID [0],  
275 0342 2 .PRCNAM_DSC_ADR);  
276 0343 2  
277 0344 2 !+  
278 0345 2 ! Free the event flag.  
279 0346 2 !-  
280 0347 2  
281 0348 2 LIB$FREE_EF (EVENT_FLAG);  
282 0349 2  
283 0350 2 !+  
284 0351 2 ! Check for errors.  
285 0352 2 !-  
286 0353 2  
287 0354 2 IF NOT .JPI_STATUS  
288 0355 2 THEN  
289 0356 2 RETURN .JPI_STATUS;  
290 0357 2  
291 0358 2 !+  
292 0359 2 ! Copy the numeric value, if desired.  
293 0360 2 !-  
294 0361 2  
295 0362 2 IF OUT_VALUE [0] NEQA 0  
296 0363 2 THEN  
297 0364 2 BEGIN  
298 0365 2 IF .RET_TYPE GTRU LIB$K_FMT_MAXSTRING ! Is it a number?  
299 0366 2 THEN  
300 0367 2 BEGIN  
301 0368 2 OUT_VALUE [0] = .RET_NUMBER [0];  
302 0369 2 IF .RET_TYPE EQL LIB$K_FMT_DATE OR  
303 0370 2 .RET_TYPE EQL LIB$K_FMT_PRIVILEGE  
304 0371 2 THEN  
305 0372 2 OUT_VALUE [1] = .RET_NUMBER [1]; ! Store second longword  
306 0373 2 END;  
307 0374 2 END;  
308 0375 2  
309 0376 2 !+  
310 0377 2 ! Store string value if desired.  
311 0378 2 !-  
312 0379 2  
313 0380 2 IF NOT NULLPARAMETER (5)  
314 0381 2 THEN  
315 0382 2 BEGIN  
316 0383 2 COPY_STATUS = LIB$COPY_R_DX6 (.RET_LENGTH, RET_STRING,  
317 0384 2 OUT_STRING [0,0,0,0]);  
318 0385 2 IF NOT NULLPARAMETER (6)  
319 0386 2 THEN  
320 0387 2 BEGIN  
321 0388 2 !+  
322 0389 2 ! Store result string length.  
323 0390 2 !-  
324 0391 2 OUT_LEN [0] = .RET_LENGTH;  
325 0392 2 IF .COPY_STATUS EQCU LIB$_STRTRU  
326 0393 2 THEN  
327 0394 2 LIB$ANALYZE_SDESC_R2 (OUT_STRING [0,0,0,0]; OUT_LEN [0]);
```



```

328
329
330
331
332
333
334
335
336
337
0395
0396
0397
0398
0399
0400
0401
0402
0403
0404

```

```

END:
RETURN .COPY_STATUS;
END
ELSE IF .RET_TYPE LEQU LIB$K_FMT_MAXSTRING
THEN
RETURN LIB$_INVARG; ! Only string value, but nothing to return it in
RETURN SSS_NORMAL; ! Success
END;

```

! End of routine LIB\$GETJPI

```

.TITLE LIB$GETJPI Get Job/Process Information
.IDENT \1-003\

```

```

.EXTRN LIB$$GETJPI, LIB$ANALYZE_SDESC_R2
.EXTRN LIB$GET_EF, LIB$FREE_EF
.EXTRN LIB$COPY_R_DX6
.EXTRN LIB$_INVARG, LIB$_STRTRU
.EXTRN LIB$_WRONUMARG

```

.PSECT _LIB\$CODE,NOWRT, SHR, PIC,2

```

.ENTRY LIB$GETJPI, Save R2,R3,R4,R5,R6,R7
MOVAB LIB$ANALYZE_SDESC_R2, R7

```

57	00000000G	00	00FC	00000	00002	0168
5E	FDE4	CE	9E	00009		
6C		J4	83	0000E		0300
02		50	91	00012		
		08	1B	00015		
50	00000000G	8F	D0	00017		
			04	0001E		
03		6C	91	0001F	1\$:	0306
		05	1F	00022		
	OC	AC	D5	00024		
		04	12	00027		
		53	D4	00029	2\$:	0308
		1B	11	0002B		
0E	AE	010E	8F	B0	0002D	0311
	50	OC	AC	D0	00033	0314
			67	16	00037	
OC	AE		51	B0	00039	
10	AE		52	D0	0003D	
	0E		50	E9	00041	
	53	OC	AE	9E	00044	0315
		08	AE	9F	C0048	0318
00000000G	00		01	FB	0004B	0325
	01		50	E8	00052	0326
				04	00055	
			53	DD	00056	0342
		08	AC	DD	00058	0341
		10	AE	DD	0005B	
		OC	AE	9F	0005E	0334
		14	AE	9F	00061	
		28	AE	9F	00064	
		34	AE	9F	00067	
00000000G	7E	04	BC	3C	0006A	0341
	00		08	FB	0006E	

```

MOVAB LIB$ANALYZE_SDESC_R2, R7
MOVAB -540(SP), SP
SUBB3 #4, (AP), DIFF
CMPB DIFF, #2
BLEQU 1$
MOVL #LIB$_WRONUMARG, R0
RET
CMPB (AP), #3
BLSSU 2$
TSTL 12(AP)
BNEQ 3$
CLRL PRCNAM_DSC_ADR
BRB 4$
MOVW #270, LCL_PRCNAM_DSC+2
MOVL PROCESS_NAME, R0
LIB$ANALYZE_SDESC R2
MOVW R1, LCL_PRCNAM_DSC
MOVL R2, LCL_PRCNAM_DSC+4
BLBC STR_STATUS, 5$
MOVAB LCL_PRCNAM_DSC, PRCNAM_DSC_ADR
PUSHAB EVENT_FLAG
CALLS #1, LIB$GET_EF
BLBS EF_STATUS, 6$
RET
PUSHL PRCNAM_DSC_ADR
PUSHL PROCESS_ID
PUSHL EVENT_FLAG
PUSHAB RET_TYPE
PUSHAB RET_LENGTH
PUSHAB RET_NUMBER
PUSHAB RET_STRING
MOVZWL @ITEM CODE, -(SP)
CALLS #8, LIB$$GETJPI

```

	52		50	D0	00075	MOVL	R0, JPI STATUS		
		08	AE	9F	00078	PUSHAB	EVENT FLAG		0348
00000000G	00		01	FB	0007B	CALLS	#1, LIB\$FREE_EF		
	04		52	E8	00082	BLBS	JPI STATUS, 7\$		0354
	50		52	D0	00085	MOVL	JPI STATUS, R0		0356
				04	00088	RET			
	50	10	AC	D0	00089	7\$: MOVL	OUT_VALUE, R0		0362
			18	13	0008D	BEQL	9\$		
	03		6E	D1	0008F	CPL	RET_TYPE, #3		0365
			13	1B	00092	BLEQU	9\$		
	60	14	AE	D0	00094	MOVL	RET_NUMBER, (R0)		0368
	0B		6E	D1	00098	CPL	RET_TYPE, #11		0369
			05	13	0009B	BEQL	8\$		
	0C		6E	D1	0009D	CPL	RET_TYPE, #12		0370
			05	12	000A0	BNEQ	9\$		
04	A0	18	AE	D0	000A2	8\$: MOVL	RET_NUMBER+4, 4(R0)		0372
	05		6C	91	000A7	9\$: CMPB	(APT), #5		0380
			40	1F	000AA	BLSSU	11\$		
		14	AC	D5	000AC	TSTL	20(AP)		
			3B	13	000AF	BEQL	11\$		
	51	1C	AE	9E	000B1	MOVA3	RET_STRING, R1		0383
	52	14	AC	D0	000B5	MOVL	OUT_STRING, R2		0384
	50	04	AE	3C	000B9	MOVZWL	RET_LENGTH, R0		
		00000000G	00	16	000BD	JSB	LIB\$COPY_R DX6		
	53		50	D0	000C3	MOVL	R0, COPY_STATUS		
	06		6C	91	000C6	CMPB	(AP), #6		0385
			1D	1F	000C9	BLSSU	10\$		
		18	AC	D5	000CB	TSTL	24(AP)		
			18	13	000CE	BEQL	10\$		
18	BC	04	AE	B0	000D0	MOVW	RET_LENGTH, @OUT_LEN		0391
00000000G	8F		53	D1	000D5	CPL	COPY_STATUS, #LIB\$_STRTRU		0392
			0A	12	000DC	BNEQ	10\$		
	50	14	AC	D0	000DE	MOVL	OUT_STRING, R0		0394
			67	16	000E2	JSB	LIB\$ANALYZE SDESC_R2		
18	BC		51	B0	000E4	MOVW	R1, @OUT_LEN		
	50		53	D0	000E8	10\$: MOVL	COPY_STATUS, R0		0396
				04	000EB	RET			
	03		6E	D1	000EC	11\$: CPL	RET_TYPE, #3		0398
			08	1A	000EF	BGTRU	12\$		
	50	00000000G	8F	D0	000F1	MOVL	#LIB\$_INVARG, R0		0400
				04	000F8	RET			
	50		01	D0	000F9	12\$: MOVL	#1, R0		0402
				04	000FC	RET			0404

: Routine Size: 253 bytes, Routine Base: _LIB\$CODE + 0000

LIB\$GETJPI
1-003

Get Job/Process Information
LIB\$GETJPI - Get Job/Process Information

I 10
16-Sep-1984 01:01:41 VAX-11 Bliss-32 V4.0-742
14-Sep-1984 12:38:59 [LIBRTL.SRC]LIBGETJPI.B32;1

Page 9
(5)

: 339 0405 1 END
: 340 0406 1
: 341 0407 0 ELUDOM

! End of module LIB\$GETJPI

PSECT SUMMARY

Name Bytes Attributes
:_LIB\$CODE 253 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	7	0	581	00:00.7
_\$255\$DUA28:[LIBRTL.OBJ]RTLLIB.L32;1	36	6	16	8	00:00.1

COMMAND QUALIFIERS

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

: Size: 253 code + 0 data bytes
: Run Time: 00:05.0
: Elapsed Time: 00:32.7
: Lines/CPU Min: 4913
: Lexemes/CPU-Min: 17396
: Memory Used: 102 pages
: Compilation Complete

LIE
1-C

