



```

LL      IIIIII  BBBB8888  AAAAAA  SSSSSSSS  NN      NN  MM      MM  BBBB8888  XX      XX
LL      IIIIII  BBBB8888  AAAAAA  SSSSSSSS  NN      NN  MM      MM  BBBB8888  XX      XX
LL      II      BB      BB  AA      AA  SS      SS      NN      NN  MMMM  MMMM  BB      BB  XX      XX
LL      II      BB      BB  AA      AA  SS      SS      NN      NN  NNNN  NN  MM      MM  BB      BB  XX      XX
LL      II      BB      BB  AA      AA  SS      SS      NN      NN  NNNN  NN  MM      MM  BB      BB  XX      XX
LL      II      BBBB8888  AA      AA  SSSSSS  NN  NN  NN  MM      MM  BBBB8888  XX      XX
LL      II      BBBB8888  AA      AA  SSSSSS  NN  NN  NN  MM      MM  BBBB8888  XX      XX
LL      II      BB      BB  AAAAAAAAAA  SS      SS  NN      NN  NNNN  MM      MM  BB      BB  XX      XX
LL      II      BB      BB  AAAAAAAAAA  SS      SS  NN      NN  NNNN  MM      MM  BB      BB  XX      XX
LL      II      BB      BB  AA      AA  SS      SS  NN      NN  NN  MM      MM  BB      BB  XX      XX
LL      II      BB      BB  AA      AA  SS      SS  NN      NN  NN  MM      MM  BB      BB  XX      XX
LLLLLLLLLLLL  IIIIII  BBBB8888  AA      AA  SSSSSSSS  NN      NN  MM      MM  BBBB8888  XX      XX
LLLLLLLLLLLL  IIIIII  BBBB8888  AA      AA  SSSSSSSS  NN      NN  MM      MM  BBBB8888  XX      XX

```

```

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 LIB$ASN_WTH_MBX (
2 0002 0
3 0003 0 IDENT = '1-007' ! File: LIBASNMBX.B32 Edit: RKR1007
4 0004 0
5 0005 0 ) =
6 0006 1 BEGIN
7 0007 1
8 0008 1 |
9 0009 1 |*****
10 0010 1 |*
11 0011 1 |* COPYRIGHT (c) 1978, 1980, 1982, 1984 BY *
12 0012 1 |* DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS. *
13 0013 1 |* ALL RIGHTS RESERVED. *
14 0014 1 |*
15 0015 1 |* THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED *
16 0016 1 |* ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE *
17 0017 1 |* INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER *
18 0018 1 |* COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY *
19 0019 1 |* OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY *
20 0020 1 |* TRANSFERRED. *
21 0021 1 |*
22 0022 1 |* THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE *
23 0023 1 |* AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT *
24 0024 1 |* CORPORATION. *
25 0025 1 |*
26 0026 1 |* DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS *
27 0027 1 |* SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL. *
28 0028 1 |*
29 0029 1 |*
30 0030 1 |*****
31 0031 1 |
32 0032 1 |
33 0033 1 |**
34 0034 1 | FACILITY: VMS Run Time Library (LIB)
35 0035 1 |
36 0036 1 | ABSTRACT:
37 0037 1 |
38 0038 1 | This module contains a routine to create a unique mailbox and
39 0039 1 | assign a channel to a device with this mailbox assigned.
40 0040 1 |
41 0041 1 | ENVIRONMENT: VAX/VMS Operating System
42 0042 1 |
43 0043 1 | AUTHOR: Darrell Duffy , CREATION DATE: 8-November-1979
44 0044 1 |
45 0045 1 | MODIFIED BY:
46 0046 1 |
47 0047 1 | 1-001 - Original. 8-Nov-1979
48 0048 1 | 1-002 - SBL1002 - Make FAO control string PIC. Also return error
49 0049 1 | if $FAO fails. SBL 10-Dec-1979
50 0050 1 | 1-003 - Allow null second and third arguments. SBL 16-Jan-1980
51 0051 1 | 1-004 - Enhance to recognize additional classes of string descriptors
52 0052 1 | by copying the DEVNAM string into an internal buffer if
53 0053 1 | we suspect it does not "read" like a CLASS_S string
54 0054 1 | descriptor.
55 0055 1 | RKR 29-MAY-1981
56 0056 1 | 1-005 - Revise enhancement to accomplish the same thing by using
57 0057 1 | LIB$ANALYZE_SDESC_R3 to fill in a local fixed-string descr..

```

```

: 58      0058 1  !      rather than doing a copy operation. This saves dragging in
: 59      0059 1  !      LIB$SCOPY and LIB$VM if this routine is linked with
: 60      0060 1  !      /NOSYSSHR. RKR 4-SEP-1981.
: 61      0061 1  ! 1-006 - Add General Addressing for all external references. DGP 25-Sep-1981.
: 62      0062 1  ! 1-007 - Redirect jsb's from LIB$ANALYZE_SDESC_R3 to
: 63      0063 1  !      LIB$ANALYZE_SDESC_R2. RKR 18-NOV-1981.
: 64      0064 1  ! --
```

## Definitions

```
66 0065 1 %SBTTL 'Definitions'
67 0066 1
68 0067 1
69 0068 1 | SWITCHES:
70 0069 1 |
71 0070 1 |
72 0071 1 SWITCHES ADDRESSING_MODE (EXTERNAL = GENERAL, NONEXTERNAL = WORD_RELATIVE);
73 0072 1 |
74 0073 1 |
75 0074 1 | TABLE OF CONTENTS:
76 0075 1 |
77 0076 1 |
78 0077 1 |
79 0078 1 |
80 0079 1 | INCLUDE FILES:
81 0080 1 |
82 0081 1 |
83 0082 1 |     REQUIRE 'RTLIN:STRLNK'; ! for linkage to LIB$ANALYZE_SDESC_R2
84 0267 1 |     LIBRARY 'RTLSTARLE';
85 0268 1 |     REQUIRE 'RTLIN:RTLPSECT';
86 0363 1 |
87 0364 1 |
88 0365 1 | MACROS:
89 0366 1 |
90 0367 1 |
91 0368 1 |
92 0369 1 | EQUATED SYMBOLS:
93 0370 1 |
94 0371 1 |
95 0372 1 |
96 0373 1 |
97 0374 1 | PSECT DEFINITIONS:
98 0375 1 |
99 0376 1 |     DECLARE_PSECTS (LIB);
100 0377 1 |
101 0378 1 |
102 0379 1 |
103 0380 1 | OWN STORAGE:
104 0381 1 |
105 0382 1 |
106 0383 1 |
107 0384 1 |
108 0385 1 | EXTERNAL REFERENCES:
109 0386 1 |
110 0387 1 |
111 0388 1 EXTERNAL ROUTINE
112 0389 1     LIB$ANALYZE_SDESC_R2 : LIB$ANALYZE_SDESC_JSB_LINK ;
```

```
114 0390 1 XSBTTL 'LIB$ASN_WTH_MBX Assign to a Device with a Mailbox'
115 0391 1 GLOBAL ROUTINE LIB$ASN_WTH_MBX (
116 0392 1
117 0393 1     DEVNAM,      : device name
118 0394 1     MAXMSG,     : max messages in mailbox
119 0395 1     BUFQUO,     : buffer quota for mailbox
120 0396 1     DEVCHN,     : device channel
121 0397 1     MBXCHN,    : mailbox channel
122 0398 1
123 0399 1     ) =
124 0400 1
125 0401 1 ++
126 0402 1 | FUNCTIONAL DESCRIPTION:
127 0403 1 |
128 0404 1 |     This routine is used to assign a channel with a unique mailbox
129 0405 1 |     for obtaining attention messages or network messages. The
130 0406 1 |     mailbox is a unique mailbox which is not known by a logical
131 0407 1 |     name in the group logical name table. The reason that this is
132 0408 1 |     desirable is that if more than one copy of the image is
133 0409 1 |     running in the same group the mailboxes will be the same and
134 0410 1 |     the mailbox messages will interfere.
135 0411 1 |
136 0412 1 |     The mailbox is referred to by physical name MBcuu instead of
137 0413 1 |     logical name to save overhead in creating its name.
138 0414 1 |
139 0415 1 | FORMAL PARAMETERS:
140 0416 1 |
141 0417 1 |     DEVNAM.rt.dx  Address of a descriptor of the device name
142 0418 1 |     MAXMSG.rl.r   Address of longword value of max messages in
143 0419 1 |                   mailbox
144 0420 1 |                   If omitted, the system supplies a default value
145 0421 1 |     BUFQUO.rl.r   Address of longword value of buffer quota for
146 0422 1 |                   mailbox
147 0423 1 |                   If omitted, the system supplies a default value
148 0424 1 |     DEVCHN.wv.r   Address to return the device channel as a word
149 0425 1 |     MBXCHN.wv.r   Address to return the mailbox channel as a word
150 0426 1 |
151 0427 1 | IMPLICIT INPUTS:
152 0428 1 |
153 0429 1 |     NONE
154 0430 1 |
155 0431 1 | IMPLICIT OUTPUTS:
156 0432 1 |
157 0433 1 |     NONE
158 0434 1 |
159 0435 1 | ROUTINE VALUE:
160 0436 1 | COMPLETION CODES:
161 0437 1 |
162 0438 1 |     Status from the system services
163 0439 1 |     or from LIB$ANALYZE_SDESC_R2 if it fails.
164 0440 1 |
165 0441 1 | SIDE EFFECTS:
166 0442 1 |
167 0443 1 |     NONE
168 0444 1 |
169 0445 1 | --
```

```
171 0446 1
172 0447 2 BEGIN
173 0448 2
174 0449 2 LITERAL
175 0450 2 MBXSIZ = 10 ; ! Max size of a mailbox name
176 0451 2
177 0452 2 LOCAL
178 0453 2 STATUS, ! Service status
179 0454 2 CHNCHAR : BLOCK [DIB$K_LENGTH, BYTE], ! Channel characteristics
180 0455 2 DIBDSC : VECTOR [2], ! Descriptor of CHNCHAR
181 0456 2 FAODSC : BLOCK [8, BYTE], ! FAO control string descriptor
182 0457 2 LOC_FIX_DESC : BLOCK [8, BYTE], ! Local fixed-string descriptor
183 0458 2 MBXBUF : VECTOR [MBXSIZ, BYTE], ! Buffer to build mailbox name
184 0459 2 MBXLST : VECTOR [2], ! FAO list for mailbox name
185 0460 2 MBXDSC : VECTOR [2], ! Descriptor of mailbox name
186 0461 2 buffer
187 0462 2 PTR; ! General pointer
188 0463 2
189 0464 2 MAP
190 0465 2 DEVNAM : REF BLOCK [, BYTE] ; ! Device name descriptor
191 0466 2
192 0467 2 BUILTIN
193 0468 2 NULLPARAMETER;
194 0469 2
195 0470 2 !+
196 0471 2 First create the mailbox, with only owner protection
197 0472 2 !-
198 0473 2
199 P 0474 2 STATUS = $CREMBX ( ! Create a mailbox
200 P 0475 2 CHAN = .MBXCHN, ! Return channel to caller
201 P 0476 2 MAXMSG = ( ! Maximum number of messages
202 P 0477 2 IF NULLPARAMETER (2)
203 P 0478 2 THEN 0
204 P 0479 2 ELSE ..MAXMSG),
205 P 0480 2 BUFQUO = ( ! Buffer quota
206 P 0481 2 IF NULLPARAMETER (3)
207 P 0482 2 THEN 0
208 P 0483 2 ELSE ..BUFQUO),
209 P 0484 2 PROMSK = 'X'FF00', ! System and Owner only RWED
210 P 0485 2 LOGNAM = 0 ! No logical name
211 0486 2 );
212 0487 2
213 0488 2 IF NOT .STATUS ! Return any error status here
214 0489 2 THEN RETURN .STATUS ;
215 0490 2
216 0491 2 DIBDSC [0] = DIB$K_LENGTH; ! Build descriptor for char buffer
217 0492 2 DIBDSC [1] = CHNCHAR;
218 0493 2
219 P 0494 2 STATUS = $GETCHN ( ! Obtain the mailbox name
220 P 0495 2 CHAN = (.MBXCHN) <0, 16, 0>, ! Mailbox channel
221 P 0496 2 PRIBUF = DIBDSC ! Primary char buffer
222 0497 2 );
223 0498 2
224 0499 2 IF NOT .STATUS ! Return service status if error
225 0500 2 THEN RETURN .STATUS ;
226 0501 2
```

```
228 0502 2
229 0503 2
230 0504 2 Build parameters for FAO
231 0505 2
232 0506 2
233 0507 2 PTR = .CHNCHAR [DIB$W_DEVNAMOFF]; ! Offset to the name
234 0508 2
235 0509 2 IF .PTR EQL 0 ! No device name returned?
236 0510 2 THEN RETURN STATUS = $$$_IVDEVNAM ; ! Return invalid device name
237 0511 2
238 0512 2 MBXLST [0] = CHNCHAR + .PTR; ! Data list has pointer to the
239 0513 2 ! name
240 0514 2 MBXLST [1] = .CHNCHAR [DIB$W_UNIT]; ! The unit number to convert
241 0515 2
242 0516 2 MBXDSC [0] = MBXSIZ; ! buffer size
243 0517 2 MBXDSC [1] = MBXBUF; ! Address of the buffer
244 0518 2
245 0519 2 !+
246 0520 2 ! Build FAO control string.
247 0521 2 !-
248 0522 2
249 0523 2 FAODSC [DSC$B_CLASS] = DSC$K_CLASS_S; ! Static class
250 0524 2 FAODSC [DSC$B_DTYPE] = DSC$K_DTYPE_T; ! Text string
251 0525 2 FAODSC [DSC$W_LENGTH] = %CHARCOUNT ('!AC!UW:');
252 0526 2 FAODSC [DSC$A_POINTER] = UPLIT BYTE ('!AC!UW:');
253 0527 2
254 P 0528 2 STATUS = $FAOL ( ! Build the whole mailbox name
255 P 0529 2 CTRSTR = FAODSC, ! Control string descriptor
256 P 0530 2 OUTLEN = MBXDSC [0], ! Length goes back in descriptor
257 P 0531 2 OUTBUF = MBXDSC, ! Descriptor is here
258 P 0532 2 PRMLST = MBXLST ! Data list is here
259 0533 2 );
260 0534 2
261 0535 2 IF NOT .STATUS
262 0536 2 THEN RETURN .STATUS; ! Return FAO error code
263 0537 2
264 0538 2 !+
265 0539 2 ! If it cannot be readily determined that the device name string
266 0540 2 ! descriptor "reads" like a CLASS_S descriptor, we construct a local
267 0541 2 ! fixed-length string descriptor to point to the "new" class of string.
268 0542 2 ! The length and address fields in the descriptor are filled with info
269 0543 2 ! returned by calling LIB$ANALYZE_SDESC_R2.
270 0544 2 !-
271 0545 2 IF .DEVNAM [DSC$B_CLASS] GTRU DSC$K_CLASS_D
272 0546 2 THEN
273 0547 2 BEGIN
274 0548 2 LOCAL
275 0549 2 STAT; ! status from LIB$ANALYZE_SDESC_R2
276 0550 2
277 0551 2 !+
278 0552 2 ! Get length and address of DEVNAM string and put in descriptor.
279 0553 2 !-
280 0554 2 STAT = LIB$ANALYZE_SDESC_R2 ( .DEVNAM ;
281 0555 2 LOC_FIX_DESC [DSC$W_LENGTH],
282 0556 2 LOC_FIX_DESC [DSC$A_POINTER] ) ;
283 0557 2
284 0558 2 !+
! If string class unrecognized, quit.
```



```

285 0559      !-
286 0560      IF NOT .STAT THEN RETURN .STAT ;
287 0561
288 0562      !+
289 0563      !- Fill in rest of fixed-length string descriptor
290 0564
291 0565      LOC_FIX_DESC [DSC$B_CLASS] = DSC$K_CLASS_S ;
292 0566      LOC_FIX_DESC [DSC$B_DTYPE] = DSC$K_DTYPE_T ;
293 0567
294 0568      END;
295 0569
296 0570      !+
297 0571      !- Now do the assign
298 0572
299 P 0573      STATUS = $ASSIGN (
300 P 0574          DEVMAM = (IF .DEVMAM [DSC$B_CLASS] GTRU DSC$K_CLASS_D
301 P 0575              THEN LOC_FIX_DESC ! use local descriptor
302 P 0576              ELSE .DEVNAM), ! else use caller's
303 P 0577          CHAN = .DEVCHN, ! Place to put channel number
304 P 0578          MBXNAM = MBXDSC ! Name we built with FAO
305 0579          );
306 0580
307 0581      RETURN .STATUS ! Return final status
308 0582
309 0583      END;

```

```

                                .TITLE LIB$ASN_WTH_MBX
                                .IDENT  \1-007\
                                .PSECT  _LIB$CODE,NOWRT, SHR, PIC,2
3A 57 55 21 43 41 21 5F 0000 P.AAA: .ASCII  \_!AC!UW:\
                                .EXTRN  LIB$ANALYZE_SDESC_R2
                                .EXTRN  SYSS$CREMBX, SYSS$GETCHN
                                .EXTRN  SYSS$FAOL, SYSS$ASSIGN

                                .ENTRY  LIB$ASN_WTH_MBX, Save R2,R3,R4
SE  FF58 CE 001C 00000 MOVAB -168(SP), SP
      7E 7C 00007 CLRQ -(SP)
7E  FF00 8F 3C 00009 MOVZWL #65280, -(SP)
03  6C 91 0000E CMPB (AP), #3
      05 1F 00011 BLSSU 1$
      0C AC D5 00013 TSTL 12(AP)
      04 12 00016 BNEQ 2$
      7E D4 00018 1$: CLRL -(SP)
      03 11 0001A BRB 3$
      0C BC DD 0C01C 2$: PUSHL @BUFQUO
02  6C 91 0001F 3$: CMPB (AP), #2
      05 1F 00022 BLSSU 4$
      08 AC D5 00024 TSTL 8(AP)
      04 12 00027 BNEQ 5$
      7E D4 00029 4$: CLRL -(SP)
      03 11 0002B BRB 6$
      08 BC DD 0002D 5$: PUSHL @MAXMSG
      14 AC DD 00030 6$: PUSHL MBXCHN

```

0391  
0486

00000000G	00		7E	D4	00033	CLRL	-(SP)			
	54		07	FB	00035	CALLS	#7, SYSS\$CREMBX			
	6A		50	D0	0003C	MOVL	R0, STATUS			
	2C	74	54	E9	0003F	BLBC	STATUS, 8\$	0488		
	30	AE	8F	9A	00042	MOVZBL	#116, DIBDSC	0491		
		AE	34	9E	00047	MOVAB	CHNCHAR, DIBDSC+4	0492		
			7E	7C	0004C	CLRG	-(SP)	0497		
			34	AE	9F	PUSHAB	DIBDSC			
			7E	D4	00051	CLRL	-(SP)			
00000000G	7E	14	BC	3C	00053	MOVZWL	@MBXCHN, -(SP)			
	00		05	FB	00057	CALLS	#5, SYSS\$GETCHN			
	54		50	D0	0005E	MOVL	R0, STATUS			
	48		54	E9	00061	BLBC	STATUS, 8\$	0499		
	50	42	AE	3C	00064	MOVZWL	CHNCHAR+14, PTR	0507		
			0B	12	00068	BNEQ	7\$	0509		
	54	0144	8F	3C	0006A	MOVZWL	#324, STATUS	0510		
	50	0144	8F	3C	0006F	MOVZWL	#324, R0			
			04	00074		RET				
	08	AE	34	AE40	9E	00075	7\$: MOVAB	CHNCHAR[PTR], MBXLST	0512	
	0C	AE	40	AE	3C	0007B	MOVZWL	CHNCHAR+12, MBXLST+4	0514	
		6E		0A	D0	00080	MOVL	#10, MBXDSC	0516	
	04	AE	10	AE	9E	00083	MOVAB	MBXBUF, MBXDSC+4	0517	
	24	AE	010E	0008	8F	D0	00088	MOVL	#17694728, FAODSC	0525
	28	AE	FF64	CF	9E	00090	MOVAB	P.AAA, FAODSC+4	0526	
			08	AE	9F	00096	PUSHAB	MBXLST	0533	
			04	AE	9F	00099	PUSHAB	MBXDSC		
			08	AE	9F	0009C	PUSHAB	MBXDSC		
			30	AE	9F	0009F	PUSHAB	FAODSC		
00000000G	00		04	FB	000A2	CALLS	#4, SYSS\$FAOL			
	54		50	D0	000A9	MOVL	R0, STATUS			
	45		54	E9	000AC	8\$: BLBC	STATUS, 12\$	0535		
	53	04	AC	D0	000AF	MOVL	DEVNAM, R3	0545		
	02	03	A3	91	000B3	CMPB	3(R3), #2			
			1A	1B	000B7	BLEQU	9\$			
	50		53	D0	000C7	MOVL	R3, R0	0555		
		00000000G	00	16	000BC	JSB	LIB\$ANALYZE_SDESC_R2			
	1C	AE	51	B0	000C2	MOVW	R1, LOC_FIX_DESC			
	20	AE	52	D0	000C6	MOVL	R2, LOC_FIX_DESC+4	0556		
		2A	50	E9	000CA	BLBC	STAT, 13\$	0560		
	1E	AE	010E	8F	B0	000CD	MOVW	#270, LOC_FIX_DESC+2	0566	
			5E	DD	000D3	9\$: PUSHL	SP	0579		
			7E	D4	000D5	CLRL	-(SP)			
		10	AC	DD	000D7	PUSHL	DEVCHN			
	02	03	A3	91	000DA	CMPB	3(R3), #2			
			08	1B	000DE	BLEQU	10\$			
	50	28	AE	9E	000E0	MOVAB	LOC_FIX_DESC, R0			
			50	DD	000E4	PUSHL	R0			
			02	11	000E6	BRB	11\$			
			53	DD	000E8	10\$: PUSHL	R3			
00000000G	00		04	FB	000EA	11\$: CALLS	#4, SYSS\$ASSIGN			
	54		50	D0	000F1	MOVL	R0, STATUS			
	50		54	D0	000F4	12\$: MOVL	STATUS, R0	0581		
			04	000F7	13\$: RET			0583		

; Routine Size: 248 bytes, Routine Base: \_LIB\$CODE + 0008

```

: 310      0584 1 END
: 311      0585 0 ELUDOM
:                               !End of module
  
```

PSECT SUMMARY

Name	Bytes	Attributes
_LIB\$CODE	256	NOVEC,NOWRT, RD , EXE, SHR, LCL, REL, CON, PIC,ALIGN(2)

Library Statistics

File	----- Symbols -----		Pages Mapped	Processing Time
	Total	Loaded		
_\$255\$DUA28:[SYSLIB]STARLET.L32;1	9776	17	0	581 00:00.8

COMMAND QUALIFIERS

```

BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/NOTRACE/LIS=LIS$:LIBASNMBX/OBJ=OBJ$:LIBASNMBX MSRCS$:LIBASNMBX/UPDATE=(ENHS$:LIBASNMBX)
  
```

```

: Size:          248 code + 8 data bytes
: Run Time:      00:05.8
: Elapsed Time: 00:28.2
: Lines/CPU Min: 6051
: Lexemes/CPU-Min: 37148
: Memory Used: 103 pages
: Compilation Complete
  
```

RTLMACB32 REQ	STRMACROS REQ	RTLOOBBG REQ	RTLPSECT REQ	STRLNK REQ	RTLMACMAR MAR	LIBASCEBC LIS	LIBASTINP LIS	LIBBINTRE LIS	LIBCHAR LIS
					LIBDEF FOR	LIBANASTR LIS	LIBASNMBX LIS	LIBBBCCI LIS	
					LIBABUPCA LIS	LIBADDP LIS			
					SIGDEF FOR		LIBASCTIM LIS		
							LIBATTACH LIS		
					LIBTABMAC MAR	LIBA2EREV LIS	LIBBBSST LIS	LIBCALLG LIS	LIBCLICAL LIS
						LIBADDP LIS			