

[illegible]

STB
1-(

.....

```

SSSSSSSS TTTTTTTTTT RRRRRRRR FFFFFFFF IIIIII NN NN DDDDDDDD FFFFFFFF IIIIII
SSSSSSSS TTTTTTTTTT RRRRRRRR FFFFFFFF IIIIII NN NN DDDDDDDD FFFFFFFF IIIIII
SS      TT      RR      RR      FF      II      NN      NN      DD      DD      FF
SS      TT      RR      RR      FF      II      NN      NN      DD      DD      FF
SS      TT      RR      RR      FF      II      NNNN      NN      DD      DD      FF
SS      TT      RR      RR      FF      II      NNNN      NN      DD      DD      FF
      SSSSSS      TTT      RRRRRRRR      FFFFFFFF      II      NN      NN      DD      DD      FFFFFFFF
      SSSSSS      TTT      RRRRRRRR      FFFFFFFF      II      NN      NN      DD      DD      FFFFFFFF
      SS      TT      RR      RR      FF      II      NN      NN      DD      DD      FF
      SS      TT      RR      RR      FF      II      NN      NNNN      DD      DD      FF
      SS      TT      RR      RR      FF      II      NN      NNNN      DD      DD      FF
      SS      TT      RR      RR      FF      II      NN      NN      DD      DD      FF
SSSSSSSS TT      RR      RR      FF      IIIIII NN      NN      DDDDDDDD FF      IIIIII
SSSSSSSS TT      RR      RR      FF      IIIIII NN      NN      DDDDDDDD FF      IIIIII

LL      IIIIII      SSSSSSSS
LL      IIIIII      SSSSSSSS
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 STR$FIND_FIRST ( ! Find first character in or not in set
2 0002 0
3 0003 0 IDENT = '1-002' ! File: STR$FINDFI.B32 Edit: RKR1002
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: String support library
35 0035 1
36 0036 1 ABSTRACT: This module contains procedures to find the first character
37 0037 1 in a set or not in a set.
38 0038 1
39 0039 1 ENVIRONMENT: User mode, AST reentrant
40 0040 1
41 0041 1 AUTHOR: R. Reichert, CREATION DATE: 26-July-1982
42 0042 1
43 0043 1 MODIFIED BY:
44 0044 1
45 0045 1 1-001 - Original. RKR 19-APR-1982.
46 0046 1 1-002 - Code improvements. RKR 24-Nov-1982.
47 0047 1 --
48 0048 1 !<BLF/PAGE>
```

```
50 0049 1 |
51 0050 1 | SWITCHES:
52 0051 1 |
53 0052 1 |
54 0053 1 | SWITCHES ADDRESSING_MODE
55 0054 1 | (EXTERNAL = GENERAL, NONEXTERNAL = WORD_RELATIVE);
56 0055 1 |
57 0056 1 |
58 0057 1 | LINKAGES
59 0058 1 |
60 0059 1 | REQUIRE 'RTLIN:STRLNK'; ! Linkage to STR$ANALYZE_SDESC_R1
61 0244 1 |
62 0245 1 | TABLE OF CONTENTS:
63 0246 1 |
64 0247 1 |
65 0248 1 | FORWARD ROUTINE
66 0249 1 | STR$FIND_FIRST_IN_SET, ! Find first occurrence of character
67 0250 1 | ! in a set of characters
68 0251 1 | STR$FIND_FIRST_NOT_IN_SET ; ! Find first occurrence of a character
69 0252 1 | ! not in a set of characters.
70 0253 1 |
71 0254 1 |
72 0255 1 | INCLUDE FILES:
73 0256 1 |
74 0257 1 |
75 0258 1 | REQUIRE 'RTLIN:RTLPSECT'; ! Declare PSECTS code
76 0353 1 | REQUIRE 'RTLIN:STRMACROS'; ! use string macros to write code
77 1269 1 | LIBRARY 'RTLSTARLE'; ! STARLET library for macros and symbols
78 1270 1 |
79 1271 1 |
80 1272 1 | MACROS: NONE
81 1273 1 |
82 1274 1 |
83 1275 1 |
84 1276 1 | EQUATED SYMBOLS: NONE
85 1277 1 |
86 1278 1 |
87 1279 1 |
88 1280 1 | PSECT DECLARATIONS
89 1281 1 |
90 1282 1 |
91 1283 1 | DECLARE_PSECTS (STR);
92 1284 1 |
93 1285 1 |
94 1286 1 | OWN STORAGE: NONE
95 1287 1 |
96 1288 1 |
97 1289 1 |
98 1290 1 | EXTERNAL REFERENCES
99 1291 1 |
100 1292 1 | NONE
```



```
102 1293 1 GLOBAL ROUTINE STR$FIND_FIRST_IN_SET ( ! Find first character in string
103 1294 1 ! that occurs in set
104 1295 1     STRING, ! pointer to string descriptor
105 1296 1     SET_OF_CHARS ! pointer to string descriptor which contains
106 1297 1 ! set of characters
107 1298 1
108 1299 1     ) : =
109 1300 1
110 1301 1 ++
111 1302 1 FUNCTIONAL DESCRIPTION:
112 1303 1     This routine searches STRING character by character, left to
113 1304 1     right, comparing each character selected to every character in
114 1305 1     SET_OF_CHARS. When a match is found, the process stops and
115 1306 1     INDEX is set to the position in STRING where the match was
116 1307 1     found. If no match is found, 0 is returned.
117 1308 1     If either STRING or SET_OF_CHARS is of zero length, index
118 1309 1     returned will be 0.
119 1310 1
120 1311 1 CALLING SEQUENCE:
121 1312 1
122 1313 1     INDEX.wl.v = STR$FIND_FIRST_IN_SET ( STRING1.rt.dx,
123 1314 1     SET_OF_CHARS.rt.dx )
124 1315 1
125 1316 1 FORMAL PARAMETERS:
126 1317 1
127 1318 1     STRING1.rt.dx     pointer to string descriptor
128 1319 1
129 1320 1     SET_OF_CHARS.rt.dx pointer to a string of characters which
130 1321 1     constitutes the set of characters of
131 1322 1     interest.
132 1323 1
133 1324 1 IMPLICIT INPUTS:
134 1325 1
135 1326 1     NONE
136 1327 1
137 1328 1 IMPLICIT OUTPUTS:
138 1329 1
139 1330 1     NONE
140 1331 1
141 1332 1 ROUTINE VALUE:
142 1333 1
143 1334 1     INDEX.wl.v     Position in STRING where a match was
144 1335 1     found, else 0.
145 1336 1
146 1337 1 SIDE EFFECTS:
147 1338 1
148 1339 1     May signal STR$_ILLSTRCLA on bad string class
149 1340 1
150 1341 1 --
151 1342 2 BEGIN
152 1343 2
153 1344 2 MAP STRING : REF BLOCK [ ,BYTE ],
154 1345 2 SET_OF_CHARS : REF BLOCK [ ,BYTE ];
155 1346 2
156 1347 2 BUILTIN
157 1348 2 SCANC;
158 1349 2
```

```
159 1350 2 LOCAL
160 1351 2 ONE : INITIAL (1),      ! Constant
161 1352 2 BYTES_REMAINING,      ! After SCANC
162 1353 2 MATCH_BYTE,           ! After SCANC
163 1354 2 SCAN_TABLE : VECTOR [256, BYTE], ! Table for SCANC operation
164 1355 2 STR_LEN : WORD,       ! length of STRING
165 1356 2 STR_ADDR,            ! address of string
166 1357 2 SET_LEN : WORD,       ! length of SET_OF_CHARS
167 1358 2 SET_ADDR : REF VECTOR [, BYTE]; ! address of SET_OF_CHARS
168 1359 2
169 1360 2 + Initialize SCAN_TABLE to zeroes.
170 1361 2 -
171 1362 2 CH$FILL (0, 256, SCAN_TABLE);
172 1363 2
173 1364 2 +
174 1365 2 - Extract the length and starting address of STRING and SET_OF_CHARS.
175 1366 2 Special-case zero length strings.
176 1367 2
177 1368 2 $STR$GET_LEN_ADDR ( STRING, STR_LEN, STR_ADDR ) ;
178 1369 2 $STR$GET_LEN_ADDR ( SET_OF_CHARS, SET_LEN, SET_ADDR ) ;
179 1370 2
180 1371 2 IF .STR_LEN EQL 0 OR
181 1372 2 .SET_LEN EQL 0 THEN RETURN 0 ;
182 1373 2
183 1374 2 +
184 1375 2 - Set a 1 bit in each character position in SCAN_TABLE that corresponds
185 1376 2 to a character in SET_OF_CHARS.
186 1377 2
187 1378 2 DECR I FROM .SET_LEN - 1 TO 0
188 1379 2 DO
189 1380 2 SCAN_TABLE [ .SET_ADDR [ .I ] ] = 1 ;
190 1381 2
191 1382 2 + Find out if one of the characters in SET_OF_CHARS appears in
192 1383 2 STRING.
193 1384 2 -
194 1385 2 SCANC ( STR_LEN, .STR_ADDR, SCAN_TABLE, ONE ; BYTES_REMAINING,
195 1386 2 MATCH_BYTE );
196 1387 2
197 1388 2 IF .BYTES_REMAINING NEQ 0 ! If scanc ran off end of string
198 1389 2 THEN
199 1390 2 RETURN (.MATCH_BYTE - .STR_ADDR + 1);
200 1391 2
201 1392 2 RETURN 0;
202 1393 2 END ;
203 1394 1 ! Procedure STR$FIND_FIRST_IN_SET
```

00FC 00000
57 00000000G 00 9E 00002

```
.TITLE STR$FIND_FIRST
.IDENT \1-002\

.EXTRN STR$ANALYZE_SDESC_R1

.PSECT _STR$CODE,NOWRT, SHR, PIC,2

.ENTRY STR$FIND_FIRST_IN_SET, Save R2,R3,R4,R5,R6,-; 1293
R7
MOVAB STR$ANALYZE_SDESC_R1, R7
```


STR\$FIND_FIRST
1-002

M 8
16-Sep-1984 01:38:01
14-Sep-1984 12:40:06

VAX-11 Bliss-32 V4.0-742
[LIBRTL.SRC]STRFINDFI.B32;1

Page 5
(3)

0100	8F	00	5E	FF00	CE	9E	00009	MOVAB	-256(SP), SP	:	
			56		01	D0	0000E	MOVL	#1, ONE	:	1342
			6E		00	2C	00011	MOVCS	#0, (SP), #0, #256, SCAN_TABLE	:	1363
					6E		00018			:	
			50	04	AC	D0	00019	MOVL	STRING, R0	:	1369
			02	03	A0	91	0001D	CMPB	3(R0), #2	:	
					09	1A	00021	BGTRU	1\$	:	
			53		60	B0	00023	MOVW	(R0), STR_LEN	:	
			54	04	A0	D0	00026	MOVL	4(R0), STR_ADDR	:	
					05	11	0002A	BRB	2\$	:	
					67	16	0002C	1\$: JSB	STR\$ANALYZE_SDESC_R1	:	
			53		50	7D	0002E	MOVQ	R0, STR_LEN	:	
			50	08	AC	D0	00031	2\$: MOVL	SET_OF_CHARS, R0	:	1370
			02	03	A0	91	00035	CMPB	3(R0), #2	:	
					09	1A	00039	BGTRU	3\$	:	
			52		60	B0	0003B	MOVW	(R0), SET_LEN	:	
			51	04	A0	D0	0003E	MOVL	4(R0), SET_ADDR	:	
					05	11	00042	BRB	4\$	:	
					67	16	00044	3\$: JSB	STR\$ANALYZE_SDESC_R1	:	
			52		50	D0	00046	MOVL	R0, SET_LEN	:	
					53	B5	00049	4\$: TSTW	STR_LEN	:	1372
					26	13	0004B	BEQL	7\$	:	
					52	B5	0004D	TSTW	SET_LEN	:	1373
					22	13	0004F	BEQL	7\$	:	
			52		52	3C	00051	MOVZWL	SET_LEN, I	:	1381
					08	11	00054	BRB	6\$	:	
			50		6241	9A	00056	5\$: MOVZBL	(I)[SET_ADDR], R0	:	
			6E40		01	90	0005A	MOVB	#1, SCAN_TABLE[R0]	:	
			F5		52	F4	0005E	6\$: SOBGEQ	I, 5\$	:	
56		6E	64		53	2A	00061	SCANC	STR_LEN, (STR_ADDR), SCAN_TABLE, ONE	:	1386
					50	D5	00066	TSTL	BYTES_REMAINING	:	1389
					09	13	00068	BEQL	7\$	:	
			51		54	C2	0006A	SUBL2	STR_ADDR, R1	:	1391
					51	D6	0006D	INCL	R1	:	
			50		51	D0	0006F	MOVL	R1, R0	:	
						04	00072	RET		:	
					50	D4	00073	7\$: CLRL	R0	:	1394
					04	00075	RET			:	

; Routine Size: 118 bytes, Routine Base: _STR\$CODE + 0000

STR
1-0

STR
1-0
S
R
E
L
M
C

```
205 1395 1 GLOBAL ROUTINE STR$FIND_FIRST_NOT_IN_SET ( ! Find first character in
206 1396 1 ! string that does not occur
207 1397 1 ! in set
208 1398 1     STRING, ! pointer to string descriptor
209 1399 1     SET_OF_CHARS ! pointer to string descriptor which contains
210 1400 1 ! set of characters
211 1401 1
212 1402 1     ) : =
213 1403 1
214 1404 1 ++
215 1405 1 FUNCTIONAL DESCRIPTION:
216 1406 1
217 1407 1     This routine searches STRING character by character, left to
218 1408 1     right, comparing each character selected to every character in
219 1409 1     SET_OF_CHARS. When a selected character is not found in
220 1410 1     SET_OF_CHARS, the process stops and INDEX is set to the position
221 1411 1     in STRING where the non-match was found. If all characters in
222 1412 1     STRING match some character in SET_OF_CHARS, 0 is returned.
223 1413 1
224 1414 1     If STRING is if zero length, index will be 1 since anything
225 1415 1     in set (particularly 1st element) will not be found in STRING.
226 1416 1     If length of SET_OF_CHARS is zero, index will be zero since
227 1417 1     "nothing" can always be found in STRING.
228 1418 1
229 1419 1 CALLING SEQUENCE:
230 1420 1     INDEX.wl.v = STR$FIND_FIRST_NOT_IN_SET ( STRING.rt.dx,
231 1421 1     SET_OF_CHARS.rt.dx )
232 1422 1
233 1423 1 FORMAL PARAMETERS:
234 1424 1
235 1425 1     STRING.rt.dx           pointer to string descriptor
236 1426 1
237 1427 1     SET_OF_CHARS.rt.dx     pointer to a string of characters which
238 1428 1     constitutes the set of characters of
239 1429 1     interest.
240 1430 1
241 1431 1 IMPLICIT INPUTS:
242 1432 1
243 1433 1     NONE
244 1434 1
245 1435 1 IMPLICIT OUTPUTS:
246 1436 1
247 1437 1     NONE
248 1438 1
249 1439 1 ROUTINE VALUE:
250 1440 1
251 1441 1     INDEX.wl.v             Position in STRING where a non-match was
252 1442 1                             found, else 0.
253 1443 1
254 1444 1 SIDE EFFECTS:
255 1445 1
256 1446 1     May signal STR$_ILLSTRCLA on bad string class
257 1447 1
258 1448 1 --
259 1449 2 BEGIN
260 1450 2
261 1451 2 MAP STRING : REF BLOCK [ ,BYTE ],
```



```

262      1452      2      SET_OF_CHARS : REF BLOCK [ ,BYTE ];
263      1453
264      1454      BUILTIN
265      1455      SPANC;
266      1456
267      1457      LOCAL
268      1458      ONE : INITIAL (1),      ! Constant
269      1459      BYTES_REMAINING,      ! After SPANC
270      1460      NON_MATCH_BYTE,      ! After SPANC
271      1461      SPAN_TABLE : VECTOR [256, BYTE], ! Table for SPANC operation
272      1462      STR_LEN : WORD,      ! length of STRING
273      1463      STR_ADDR,      ! address of string
274      1464      SET_LEN : WORD,      ! length of SET_OF_CHARS
275      1465      SET_ADDR : REF VECTOR [ ,BYTE]; ! address of SET_OF_CHARS
276      1466
277      1467      !+ Initialize SPAN_TABLE to zeroes.
278      1468      !-
279      1469      CH$FILL (0, 256, SPAN_TABLE);
280      1470
281      1471
282      1472      !+ Extract the length and starting address of STRING and SET_OF_CHARS.
283      1473      !- Special case zero length strings.
284      1474
285      1475      $STR$GET_LEN_ADDR ( STRING, STR_LEN, STR_ADDR );
286      1476      $STR$GET_LEN_ADDR ( SET_OF_CHARS, SET_LEN, SET_ADDR );
287      1477
288      1478      IF .SET_LEN EQL 0 THEN RETURN 0 ;
289      1479      IF .STR_LEN EQL 0 THEN RETURN 1 ;
290      1480
291      1481
292      1482      !+ Set a 1 bit in each character position in SPAN_TABLE that corresponds
293      1483      !- to a character in SET_OF_CHARS.
294      1484
295      1485      DECR I FROM .SET_LEN - 1 TO 0
296      1486      DO
297      1487      SPAN_TABLE [ .SET_ADDR [ .I ] ] = 1 ;
298      1488
299      1489      !+ Find out if one of the characters in SET_OF_CHARS does not appear in
300      1490      !- STRING.
301      1491
302      1492      SPANC ( STR_LEN, .STR_ADDR, SPAN_TABLE, ONE ; BYTES_REMAINING,
303      1493      NON_MATCH_BYTE );
304      1494
305      1495      IF .BYTES_REMAINING NEQ 0      ! If span ran off end of string
306      1496      THEN
307      1497      RETURN ( .NON_MATCH_BYTE - .STR_ADDR + 1 );
308      1498
309      1499
310      1500      RETURN 0;
311      1501      END ;
                                     ! Procedure STR$FIND_FIRST_NOT_IN_SET
```

00FC 00000

.ENTRY STR\$FIND_FIRST_NOT_IN_SET, Save R2,R3,R4,- ; 1395
R5,R6,R7

STR\$FIND_FIRST
1-002

C 9
16-Sep-1984 01:38:01
14-Sep-1984 12:40:06

VAX-11 Bliss-32 V4.0-742
[LIBRTL.SRC]STRFINDFI.B32;1

Page 8
(4)

0100 8F 00

57	00000000G	00	9E	00002	MOVAB	STR\$ANALYZE_SDESC_R1, R7	:
5E	FF00	CE	9E	00009	MOVAB	-256(SP), SP	:
56		01	D0	0000E	MOVL	#1, ONE	1449
6E		00	2C	00011	MOVCS	#0, (SP), #0, #256, SPAN_TABLE	1470
		6E		00018			:
50	04	AC	D0	00019	MOVL	STRING, R0	1476
02	03	A0	91	0001D	CMPB	3(R0), #2	:
		09	1A	00021	BGTRU	1\$	:
53		60	B0	00023	MOVW	(R0), STR_LEN	:
54	04	A0	D0	00026	MOVL	4(R0), STR_ADDR	:
		05	11	0002A	BRB	2\$	:
		67	16	0002C	1\$: JSB	STR\$ANALYZE_SDESC_R1	:
53		50	7D	0002E	MOVQ	R0, STR_LEN	:
50	08	AC	D0	00031	2\$: MOVL	SET_OF_CHARS, R0	1477
02	03	A0	91	00035	CMPB	3(R0), #2	:
		09	1A	00039	BGTRU	3\$	:
52		60	B0	0003B	MOVW	(R0), SET_LEN	:
51	04	A0	D0	0003E	MOVL	4(R0), SET_ADDR	:
		05	11	00042	BRB	4\$	:
		67	16	00044	3\$: JSB	STR\$ANALYZE_SDESC_R1	:
52		50	D0	00046	MOVL	R0, SET_LEN	:
		52	B5	00049	4\$: TSTW	SET_LEN	1479
		2A	13	0004B	BEQL	8\$	:
		53	B5	0004D	TSTW	STR_LEN	1480
		04	12	0004F	BNEQ	5\$	:
50		01	D0	00051	MOVL	#1, R0	:
			04	00054	RET		:
52		52	3C	00055	5\$: MOVZWL	SET_LEN, I	1488
		08	11	00058	BRB	7\$	:
50		6241	9A	0005A	6\$: MOVZBL	(I)[SET_ADDR], R0	:
6E40		01	90	0005E	MOVB	#1, SPAN_TABLE[R0]	:
F5		52	F4	00062	7\$: SOBGEQ	I, 6\$	:
64		53	2B	00065	SPANC	STR_LEN, (STR_ADDR), SPAN_TABLE, ONE	1493
		50	D5	0006A	TSTL	BYTES_REMAINING	1496
		09	13	0006C	BEQL	8\$	:
51		54	C2	0006E	SUBL2	STR_ADDR, R1	1498
		51	D6	00071	INCL	R1	:
50		51	D0	00073	MOVL	R1, R0	:
			04	00076	RET		:
		50	D4	00077	8\$: CLRL	R0	1501
			04	00079	RET		:

; Routine Size: 122 bytes, Routine Base: _STR\$CODE + 0076

STR\$FIND_FIRST
1-002

D 9
16-Sep-1984 01:38:01
14-Sep-1984 12:40:06

VAX-11 Bliss-32 V4.0-742
[LIBRTL.SRC]STRFINDFI.B32;1

Page 9
(5)

: 313 1502 1 END
: 314 1503 0 ELUDOM

!End of module

PSECT SUMMARY

: Name Bytes Attributes
: _STR\$CODE 240 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 4 0 581 00:00.7

COMMAND QUALIFIERS

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

: Size: 240 code + 0 data bytes
: Run Time: 00:06.9
: Elapsed Time: 00:28.8
: Lines/CPU Min: 12994
: Lexemes/CPU-Min: 33873
: Memory Used: 88 pages
: Compilation Complete

0214 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

