

BBBBBBBBBBBBBB		AAAAAAAAAA	CCCCCCCCCCCC	KKK	KKK	UUU	UUU	PPPPPPPPPPPP
BBBBBBBBBBBBBB		AAAAAAAAAA	CCCCCCCCCCCC	KKK	KKK	UUU	UUU	PPPPPPPPPPPP
BBBBBBBBBBBBBB		AAAAAAAAAA	CCCCCCCCCCCC	KKK	KKK	UUU	UUU	PPPPPPPPPPPP
BBB	BBB	AAA	AAA	CCC	KKK	UUU	UUU	PPP
BBB	BBB	AAA	AAA	CCC	KKK	UUU	UUU	PPP
BBB	BBB	AAA	AAA	CCC	KKK	UUU	UUU	PPP
BBB	BBB	AAA	AAA	CCC	KKK	UUU	UUU	PPP
BBB	BBB	AAA	AAA	CCC	KKK	UUU	UUU	PPP
BBB	BBB	AAA	AAA	CCC	KKK	UUU	UUU	PPP
BBB	BBB	AAA	AAA	CCC	KKK	UUU	UUU	PPP
BBBBBBBBBBBBBB		AAA	AAA	CCC	KKKKKKKKKK	UUU	UUU	PPPPPPPPPPPP
BBBBBBBBBBBBBB		AAA	AAA	CCC	KKKKKKKKKK	UUU	UUU	PPPPPPPPPPPP
BBBBBBBBBBBBBB		AAA	AAA	CCC	KKKKKKKKKK	UUU	UUU	PPPPPPPPPPPP
BBB	BBB	AAAAAAAAAAAAAAAA	CCC	KKK	KKK	UUU	UUU	PPP
BBB	BBB	AAAAAAAAAAAAAAAA	CCC	KKK	KKK	UUU	UUU	PPP
BBB	BBB	AAAAAAAAAAAAAAAA	CCC	KKK	KKK	UUU	UUU	PPP
BBB	BBB	AAA	AAA	CCC	KKK	UUU	UUU	PPP
BBB	BBB	AAA	AAA	CCC	KKK	UUU	UUU	PPP
BBB	BBB	AAA	AAA	CCC	KKK	UUU	UUU	PPP
BBB	BBB	AAA	AAA	CCC	KKK	UUU	UUU	PPP
BBB	BBB	AAA	AAA	CCC	KKK	UUU	UUU	PPP
BBBBBBBBBBBBBB		AAA	AAA	CCCCCCCCCCCC	KKK	KKK	UUUUUUUUUUUUUUUU	PPP
BBBBBBBBBBBBBB		AAA	AAA	CCCCCCCCCCCC	KKK	KKK	UUUUUUUUUUUUUUUU	PPP
BBBBBBBBBBBBBB		AAA	AAA	CCCCCCCCCCCC	KKK	KKK	UUUUUUUUUUUUUUUU	PPP

CCCCCCCC	RRRRRRRR	EEEEEEEEEE	AAAAAA	TTTTTTTTTT	EEEEEEEEEE	DDDDDDDD	IIIIII	RRRRRRRR
CCCCCCCC	RRRRRRRR	EEEEEEEEEE	AAAAAA	TTTTTTTTTT	EEEEEEEEEE	DDDDDDDD	IIIIII	RRRRRRRR
CC	RR	EE	AA	TT	EE	DD	II	RR
CC	RR	EE	AA	TT	EE	DD	II	RR
CC	RR	EE	AA	TT	EE	DD	II	RR
CC	RR	EE	AA	TT	EE	DD	II	RR
CC	RRRRRRRR	EEEEEEEEEE	AA	TT	EEEEEEEEEE	DD	II	RRRRRRRR
CC	RRRRRRRR	EEEEEEEEEE	AA	TT	EEEEEEEEEE	DD	II	RRRRRRRR
CC	RR RR	EE	AAAAAAAAAA	TT	EE	DD	II	RR RR
CC	RR RR	EE	AAAAAAAAAA	TT	EE	DD	II	RR RR
CC	RR RR	EE	AA AA	TT	EE	DD	II	RR RR
CC	RR RR	EE	AA AA	TT	EE	DD	II	RR RR
CC	RR RR	EE	AA AA	TT	EE	DD	II	RR RR
CC	RR RR	EE	AA AA	TT	EE	DD	II	RR RR
CCCCCCCC	RR RR	EEEEEEEEEE	AA AA	TT	EEEEEEEEEE	DDDDDDDD	IIIIII	RR RR
CCCCCCCC	RR RR	EEEEEEEEEE	AA AA	TT	EEEEEEEEEE	DDDDDDDD	IIIIII	RR RR

```

LL          IIIII
LL          IIIII
LL          II
LL          II
LL          II
LL          II
LL          II
LL          II
LL          II
LL          II
LL          II
LL          II
LL          II
LL          II
LLLLLLLLLLL
LLLLLLLLLLL

          IIIII
          IIIII
          II
          II
          II
          II
          II
          II
          II
          II
          II
          II
          II
          II
          IIIII
          IIIII

          SSSSSSSS
          SSSSSSSS
          SS
          SS
          SS
          SS
          SSSSSS
          SSSSSS
          SS
          SS
          SS
          SS
          SSSSSSSS
          SSSSSSSS

```



```
0001 0 MODULE CREATEDIR(%TITLE 'Create directory'
0002 0 IDENT = 'V04-000'
0003 0 ) =
0004 1 BEGIN
0005 1
0006 1 *****
0007 1 *
0008 1 * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
0009 1 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
0010 1 * ALL RIGHTS RESERVED.
0011 1 *
0012 1 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
0013 1 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
0014 1 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
0015 1 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
0016 1 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
0017 1 * TRANSFERRED.
0018 1 *
0019 1 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
0020 1 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
0021 1 * CORPORATION.
0022 1 *
0023 1 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
0024 1 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
0025 1 *
0026 1 *****
0027 1
0028 1
0029 1 ++
0030 1 FACILITY:
0031 1 Backup/Restore
0032 1
0033 1 ABSTRACT:
0034 1 This module contains the routines that create directories.
0035 1
0036 1 ENVIRONMENT:
0037 1 VAX/VMS user mode.
0038 1
0039 1 AUTHOR: M. Jack, CREATION DATE: 28-Sep-1980
0040 1
0041 1 MODIFIED BY:
0042 1
0043 1 V03-006 ACG0373 Andrew C. Goldstein, 11-Nov-1983 15:46
0044 1 Fix directory creation path in /INCREMENTAL case;
0045 1 allow directory creation over dangling entries.
0046 1
0047 1 V03-005 KBT0586 Keith B. Thompson 15-Aug-1983
0048 1 Remove ref. to LIB$EXTRACT_CONCEAL and change accordingly
0049 1
0050 1 V03-004 MLJ0112 Martin L. Jack, 19-Jul-1983 16:51
0051 1 Fix uninitialized backup date in /INCREMENTAL case.
0052 1
0053 1 V03-003 ACG0313 Andrew C. Goldstein, 12-Feb-1983 16:09
0054 1 Add routine subtitles
0055 1
0056 1 V03-002 MLJ53474 Martin L. Jack, 4-Feb-1983 0:15
0057 1 Track ACG0275 in LIB$CREATE_DIR.
```


CREATEDIR
V04-000

Create directory

J 16
15-Sep-1984 23:52:12
14-Sep-1984 11:53:50

VAX-11 Bliss-32 V4.0-742
[BACKUP.SRC]CREATEDIR.B32;1

Page 2
(1)

:	58	0058	1	:	
:	59	0059	1	:	
:	60	0060	1	:	V03-001 MLJ0101 Martin L. Jack, 13-Nov-1982 19:24
:	61	0061	1	:	Fix ADDRESSING_MODE(GENERAL) errors.
:	62	0062	1	:	
:	63	0063	1	:	V02-003 MLJ0063 Martin L. Jack, 31-Dec-1981 11:05
:	64	0064	1	:	Readapt from new LIB\$CREATE_DIR to handle root directories.
:	65	0065	1	:	
:	66	0066	1	:	V02-002 MLJ0062 Martin L. Jack, 3-Dec-1981 12:30
:	67	0067	1	:	Set backup date of created directory, return file ID and
:	68	0068	1	:	backup date of directory, and consider lookups that return a
:	69	0069	1	:	file ID of all ones to be nonexistent, all to support
:	70	0070	1	:	/INCREMENTAL.
:	71	0071	1	:	
:	72	0072	1	:	V02-001 MLJ0036 Martin L. Jack, 28-Aug-1981 17:32
:	73	0073	1	:	Make use of new NAM block descriptors.
:	74	0074	1	:	--

CREATEDIR
V04-000

Create directory

K 16
15-Sep-1984 23:52:12
14-Sep-1984 11:53:50

VAX-11 Bliss-32 V4.0-742
[BACKUP.SRC]CREATEDIR.B32;1

Page 3
(2)

```
: 76      0075 1 REQUIRE 'SRC$:COMMON';
: 77      1181 1 LIBRARY 'SYSS$LIBRARY:LIB';
: 78      1182 1
: 79      1183 1
: 80      1184 1 FORWARD ROUTINE
: 81      1185 1      CREATE_DIR;
: 82      1186 1
: 83      1187 1
: 84      1188 1 EXTERNAL ROUTINE
: 85      1189 1      LIB$CVT_OTB:
: 86      1190 1
: 87      1191 1
: 88      1192 1 EXTERNAL LITERAL
: 89      1193 1      LIB$_INVFILSPE;
: 90      1194 1
: 91      1195 1
: 92      1196 1 G$DEFINE();
```

! Create directory file

ADDRESSING_MODE(GENERAL);
! Convert octal to binary
! Completion status codes
! Invalid file specification

! Define common area

CREATEDIR
V04-000

Create directory
CREATE_DIR - create a directory file

L 16
15-Sep-1984 23:52:12
14-Sep-1984 11:53:50

VAX-11 Bliss-32 V4.0-742
[BACKUP.SRC]CREATEDIR.B32;1

Page 4
(3)

```

94 1197 1 %SBTTL 'CREATE_DIR - create a directory file'
95 1198 1 GLOBAL ROUTINE CREATE_DIR (
96 1199 1     DEV_DIR_SPEC: REF BBLOCK,
97 1200 1     OWNER_UIC,
98 1201 1     PROT_VALUE,
99 1202 1     MAX_VERSIONS,
100 1203 1     FILE_ID: REF BBLOCK,
101 1204 1     BAKDATE: REF VECTOR
102 1205 1 ) =
103 1206 1
104 1207 1 ++
105 1208 1 FUNCTIONAL DESCRIPTION:
106 1209 1
107 1210 1     This routine creates directory files.
108 1211 1
109 1212 1 FORMAL PARAMETERS:
110 1213 1
111 1214 1     DEV_DIR_SPEC    Address of a descriptor for the device and directory
112 1215 1                  specification. This string is a standard RMS file
113 1216 1                  specification; it must not contain a node name, file
114 1217 1                  name, file type, file version, or wild card characters;
115 1218 1                  it must contain an explicit directory; it must
116 1219 1                  reference a disk device. The string must be no longer
117 1220 1                  than 255 characters.
118 1221 1
119 1222 1     OWNER_UIC        A longword passed by immediate value that specifies the
120 1223 1                  owner UIC of the created directories. If zero, the
121 1224 1                  owner UIC is that of the parent directory.
122 1225 1
123 1226 1     PROT_VALUE       A word passed by immediate value that specifies the
124 1227 1                  file protection. If all ones, the file protection is
125 1228 1                  that of the parent directory with delete access dropped
126 1229 1                  for all access categories.
127 1230 1
128 1231 1     MAX_VERSIONS     A word passed by immediate value that specifies the
129 1232 1                  default maximum number of versions for files cataloged
130 1233 1                  in the created directories.
131 1234 1
132 1235 1     FILE_ID          The address of an array of three words that receives
133 1236 1                  the file ID of the lowest level directory.
134 1237 1
135 1238 1     BAKDATE          The address of a quadword that receives the backup date
136 1239 1                  of the lowest level directory.
137 1240 1
138 1241 1 IMPLICIT INPUTS:
139 1242 1
140 1243 1     OUTPUT_CHAN      Assigned to the output volume.
141 1244 1     QUAL             Qualifiers.
142 1245 1     COM_O_BSRDATE    Date of backup.
143 1246 1
144 1247 1 IMPLICIT OUTPUTS:
145 1248 1
146 1249 1     NONE
147 1250 1
148 1251 1 COMPLETION STATUS:
149 1252 1
150 1253 1     $$$_NORMAL       Normal successful completion; all specified directories
                        already exist
```



```
151 1254 1 |
152 1255 1 | SSS_CREATED Normal successful completion; one or more directories
153 1256 1 | created
154 1257 1 |
155 1258 1 | LIB$_INVFILSPE File specification did not contain an explicit
156 1259 1 | directory or contained a node name, file name, file
157 1260 1 | type, file version, or wildcard, or device not a disk
158 1261 1 |
159 1262 1 | $PARSE errors
160 1263 1 | $QIO errors
161 1264 1 |
162 1265 1 | SIDE EFFECTS:
163 1266 1 |
164 1267 1 | Directory files created as requested.
165 1268 1 |
166 1269 1 | --
167 1270 1 |
168 1271 2 | BEGIN
169 1272 2 | LOCAL
170 1273 2 | FAB: $FAB_DECL, ! FAB for $PARSE
171 1274 2 | NAM: $NAM_DECL, ! NAM block for $PARSE
172 1275 2 | ESA_BUFFER: VECTOR[NAM$C_MAXRSS,BYTE], ! Expanded string area
173 1276 2 | NAME_BUFFER: VECTOR[39+6,BYTE], ! Directory name buffer
174 1277 2 | NAME_DESC: BLOCK[DSC$K_2_BLN,BYTE], ! Directory name descriptor
175 1278 2 | RECATTR: BLOCK[ATR$$_RECATTR,BYTE]
176 1279 2 | VOLATILE, ! Record attributes
177 1280 2 | UCHAR: BLOCK[ATR$$_UCHAR,BYTE]
178 1281 2 | VOLATILE, ! File characteristics
179 1282 2 | FPRO: BLOCK[ATR$$_FPRO,BYTE]
180 1283 2 | VOLATILE, ! File protection
181 1284 2 | UIC: BLOCK[ATR$$_UIC,BYTE]
182 1285 2 | VOLATILE, ! File owner UIC
183 1286 2 | BAK: VECTOR[2]
184 1287 2 | VOLATILE, ! File backup date
185 1288 2 | HEADER: BLOCK[ATR$$_HEADER,BYTE]
186 1289 2 | VOLATILE, ! File header
187 1290 2 | FIB: BLOCK[FIB$$_LENGTH,BYTE], ! FIB
188 1291 2 | FIB_DESC: VECTOR[2], ! Descriptor for FIB
189 1292 2 | ATR: BLOCKVECTOR[7,8,BYTE], ! Attribute descriptors
190 1293 2 | IOSB: VECTOR[4,WORD], ! I/O status block
191 1294 2 | GROUP, ! Binary group number
192 1295 2 | MEMBER, ! Binary member number
193 1296 2 | DIR_LENGTH, ! Length of residual directory string
194 1297 2 | DIR_ADDRESS, ! Address of residual directory string
195 1298 2 | PREV_ADDRESS, ! Address of previous directory segment
196 1299 2 | REALDIR_ADDRESS, ! Temporary pointer
197 1300 2 | STATUS_2, ! Status return
198 1301 2 | STATUS_3, ! Status return
199 1302 2 | FINAL_STATUS; ! Status return
200 1303 2 | BIND
201 1304 2 | DIR_TYP_VER = UPLIT BYTE ('.DIR;1');
202 1305 2 | ! File type and version string
203 1306 2 | LABEL
204 1307 2 | PROCESS; ! Block exited when processing complete
205 1308 2 | BUILTIN
206 1309 2 | LOCC, ! LOCC instruction
207 1310 2 | ROT; ! Rotate longword
```



```
208 1311 2
209 1312 2
210 1313 2
211 1314 2
212 1315 2
213 P 1316 2 $FAB INIT(FAB=FAB,
214 P 1317 2     FNS=.DEV_DIR_SPEC[DSCSW_LENGTH],
215 P 1318 2     FNA=.DEV_DIR_SPEC[DSCSA_POINTER],
216 1319 2     NAM=NAM);
217 P 1320 2 $NAM INIT(NAM=NAM,
218 P 1321 2     NOP=<NOCONCEAL>,
219 P 1322 2     ESA=ESA_BUFFER,
220 1323 2     ESS=NAM$C_MAXRSS);
221 1324 2
222 1325 2
223 1326 2
224 1327 2
225 1328 2
226 1329 2
227 1330 2
228 1331 2 STATUS_2 = $PARSE(FAB=FAB);
229 1332 2 IF NOT .STATUS_2 AND .STATUS_2 NEQ RMS$_DNF THEN RETURN .STATUS_2;
230 1333 2
231 1334 2 ! Get the length and address of the directory string without brackets.
232 1335 2
233 1336 2 DIR_LENGTH = .NAM[NAM$B_DIR] - 2; ! Length without brackets
234 1337 2 DIR_ADDRESS = .NAM[NAM$C_DIR] + 1; ! Address without bracket
235 1338 2
236 1339 2 ! If there is a root directory locate the real directory and squish
237 1340 2 ! them together
238 1341 2
239 1342 2 IF .NAM [ NAM$V_ROOT_DIR ]
240 1343 2 THEN
241 1344 2 BEGIN
242 1345 2
243 1346 2 REGISTER
244 1347 2     R0 = 0,
245 1348 2     R1 = 1;
246 1349 2
247 1350 2 LOCAL
248 1351 2     TERMINATOR: BYTE;
249 1352 2
250 1353 2 TERMINATOR = ..NAM [ NAM$L_DIR ] + 2; ! Close = Open+2
251 1354 2
252 1355 2 IF NOT LOCC( TERMINATOR, DIR_LENGTH, .DIR_ADDRESS; R0, R1 )
253 1356 2 THEN
254 1357 2     RETURN LIB$_INVFILSPE; ! No Root Found
255 1358 2
256 1359 2 REALDIR_ADDRESS = .R1;
257 1360 2
258 1361 2 ! Found the terminator of the root directory, the real directory
259 1362 2 ! will be 2 past it, i.e. "[ROOT.][DIR]"
260 1363 2 REALDIR_ADDRESS-^
261 1364 2
262 1365 2 ! Move what is left over 2 in order to get rid of the "[[" to get:
263 1366 2 "[ROOT.DIR]"
264 1367 2 REALDIR_ADDRESS-^
```



```
265 1368 2 1
266 1369 2 CH$MOVE( .R0, .REALDIR_ADDRESS+2, .REALDIR_ADDRESS );
267 1370 2
268 1371 2 1 Adjust the direcorey length
269 1372 2
270 1373 2 DIR_LENGTH = .DIR_LENGTH - 2
271 1374 2
272 1375 2 END
273 1376 2 ELSE
274 1377 2
275 1378 2 1 If no root then the real directory is the uic directory
276 1379 2
277 1380 2 REALDIR_ADDRESS = .DIR_ADDRESS;
278 1381 2
279 1382 2 1 If the directory is in UIC format, get the octal conversion
280 1383 2
281 1384 2 IF .NAM [ NAM$V_GRP_MBR ]
282 1385 2 THEN
283 1386 2 BEGIN
284 1387 2
285 1388 2 1 Convert the group part
286 1389 2
287 1390 2 IF NOT LIB$CVT_OTB( 3, .REALDIR_ADDRESS, GROUP)
288 1391 2 THEN
289 1392 2 RETURN LIB$_INVFILSPE; ! Invalid group number
290 1393 2
291 1394 2 1 Convert the member part
292 1395 2
293 1396 2 IF NOT LIB$CVT_OTB( 3, .REALDIR_ADDRESS + 3, MEMBER)
294 1397 2 THEN
295 1398 2 RETURN LIB$_INVFILSPE ! Invalid member number
296 1399 2
297 1400 2 END;
298 1401 2
299 1402 2 1+
300 1403 2 1 Set up the FIB to look up the MFD.
301 1404 2 1-
302 1405 2
303 1406 2 CH$FILL(0, FIB$_LENGTH, FIB);
304 1407 2 FIB[FIB$_ACCTL] = FIB$_WRITE OR FIB$_NOREAD OR FIB$_NOWRITE;
305 1408 2 FIB[FIB$_NMCTL] = FIB$_SUPERSEDE;
306 1409 2 FIB[FIB$_FID_NUM] = FID$_MFD;
307 1410 2 FIB[FIB$_FID_SEQ] = FID$_MFD;
308 1411 2
309 1412 2 1+
310 1413 2 1 Set up the FIB descriptor.
311 1414 2 1-
312 1415 2
313 1416 2 FIB_DESC[0] = FIB$_LENGTH;
314 1417 2 FIB_DESC[1] = FIB;
315 1418 2
316 1419 2 1+
317 1420 2 1 Set up the name descriptor. The length is filled in as need be.
318 1421 2 1-
319 1422 2
320 1423 2 NAME_DESC[DSC$_A_POINTER] = NAME_BUFFER;
321 1424 2
```



```

322 1425 2  !+
323 1426 2  ! Beginning of block that is exited when processing is complete. FINAL_STATUS
324 1427 2  ! contains the status to be returned to caller.
325 1428 2  !-
326 1429 2  !-
327 1430 2  PROCESS: BEGIN
328 1431 3  !+
329 1432 3  ! Loop to look up directories.
330 1433 3  !-
331 1434 3  !-
332 1435 3  !-
333 1436 3  PREV ADDRESS = 0;
334 1437 3  WHILE 1 DO
335 1438 4  BEGIN
336 1439 4  !+
337 1440 4  ! Copy the file ID to FIB$W_DID so that the next lookup is done in that
338 1441 4  ! directory.
339 1442 4  !-
340 1443 4  !-
341 1444 4  !-
342 1445 4  FIB[FIB$W_DID_NUM] = .FIB[FIB$W_FID_NUM];
343 1446 4  FIB[FIB$W_DID_SEQ] = .FIB[FIB$W_FID_SEQ];
344 1447 4  FIB[FIB$W_DID_RVN] = .FIB[FIB$W_FID_RVN];
345 1448 4  !-
346 1449 4  !-
347 1450 4  !+
348 1451 4  ! Locate the next directory name.
349 1452 4  !-
350 1453 4  !-
351 1454 5  BEGIN ! block to use output registers
352 1455 5  REGISTER
353 1456 5  RO = 0;
354 1457 5  R1 = 1;
355 1458 5  LOCAL
356 1459 5  NAME_LENGTH,
357 1460 5  NAME_ADDRESS;
358 1461 5  !-
359 1462 5  NAME_ADDRESS = .DIR ADDRESS; ! Save starting point
360 1463 5  LOCCT%REF(%C'.'), DIR_LENGTH, .DIR_ADDRESS; R0, R1);
361 1464 5  NAME_LENGTH = .DIR_LENGTH - .R0; ! Length preceding dot or end
362 1465 5  DIR_ADDRESS = .R1 + 1; ! Prune to string following dot
363 1466 5  R0 = .R0 - 1;
364 1467 5  DIR_LENGTH = .R0;
365 1468 5  !-
366 1469 5  !+
367 1470 5  ! Construct the directory name concatenated with '.DIR;1' in the name
368 1471 5  ! buffer, and a descriptor for this name in the name descriptor.
369 1472 5  !-
370 1473 5  !-
371 1474 5  NAME_DESC[DSC$W_LENGTH] = .NAME_LENGTH + 6;
372 1475 5  CH$MOVE(6, DIR_TYP_VER, CH$MOVE(.NAME_LENGTH, .NAME_ADDRESS, NAME_BUFFER));
373 1476 4  END; ! block to use output registers
374 1477 4  !-
375 1478 4  !+
376 1479 4  ! Look up the directory at the current level. If the directory
377 1480 4  ! does not exist, exit the loop to begin creating directories.
378 1481 4  !-

```



```
379 1482 4
380 P 1483 4 FINAL STATUS = $QIOW(
381 P 1484 4     FONC=IOS$ ACCESS,
382 P 1485 4     CHAN=.OUTPUT_CHAN,
383 P 1486 4     IOSB=IOSB,
384 P 1487 4     P1=FIB_DESC,
385 1488 4     P2=NAME_DESC);
386 1489 4 IF .FINAL STATUS THEN FINAL STATUS = .IOSB[0];
387 1490 4 IF .FIB[FIB$W_FID_NUM] EQL 65535
388 1491 4 AND .FIB[FIB$W_FID_SEQ] EQL 65535
389 1492 4 AND .FIB[FIB$B_FID_NMX] EQL 255
390 1493 4 THEN FINAL STATUS = SSS_NOSUCHFILE;
391 1494 4 IF .FINAL STATUS EQL SSS_NOSUCHFILE
392 1495 4 THEN EXITLOOP;
393 1496 4 IF NOT .FINAL STATUS THEN LEAVE PROCESS;
394 1497 4
395 1498 4 ! Save the directory ID and previous name string in case we
396 1499 4 ! need to back out.
397 1500 4
398 1501 4
399 1502 4 PREV_ADDRESS = .DIR_ADDRESS - .NAME_DESC[DSC$W_LENGTH] + 6 - 1;
400 1503 4 FILE_ID[FID$W_NUM] = .FIB[FIB$W_DID_NUM];
401 1504 4 FILE_ID[FID$W_SEQ] = .FIB[FIB$W_DID_SEQ];
402 1505 4 FILE_ID[FID$W_RVN] = .FIB[FIB$W_DID_RVN];
403 1506 4
404 1507 4
405 1508 4 !+
406 1509 4 ! If no more directory levels were specified, all specified directories
407 1510 4 ! already exist. We do a read attributes on the last directory found
408 1511 4 ! for one of several reasons: (1) to get the attributes to propagate
409 1512 4 ! to the next directory level to be created. (2) to get the backup
410 1513 4 ! date of the directory for an incremental scan. (3) to ensure that
411 1514 4 ! the directory exists, and we haven't just hit a dangling directory
412 1515 4 ! entry.
413 1516 4 !-
414 1517 4 IF .DIR_LENGTH LEQ 0
415 1518 4 THEN
416 1519 5 BEGIN
417 1520 5     FINAL STATUS = SSS_NORMAL;
418 1521 5     EXITLOOP;
419 1522 4     END;
420 1523 3 END;
421 1524 3
422 1525 3 !+
423 1526 3 ! We have reached the level at which directories do not yet exist. FIB$W_DID
424 1527 3 ! now contains the file ID of the directory in which the new directory must be
425 1528 3 ! cataloged and the filename descriptor contains the name of the new directory.
426 1529 3 !-
427 1530 3
428 1531 3 !+
429 1532 3 ! Set up the attribute list. Because of dependencies later in the routine,
430 1533 3 ! the file header attribute must be last, preceded by the owner UIC attribute.
431 1534 3 !-
432 1535 3
433 1536 3 ATR[0, ATR$W_TYPE] = ATR$C_RECATTR; ! Record attributes
434 1537 3 ATR[0, ATR$W_SIZE] = ATR$S_RECATTR;
435 1538 3 ATR[0, ATR$L_ADDR] = RECATTR;
```



```
: 436 1539 3 ATR[1, ATR$W_TYPE] = ATR$C_UCHAR;      ! File characteristics
: 437 1540 3 ATR[1, ATR$W_SIZE] = ATR$S_UCHAR;
: 438 1541 3 ATR[1, ATR$L_ADDR] = UCHAR;
: 439 1542 3 ATR[2, ATR$W_TYPE] = ATR$C_FPRO;      ! File protection
: 440 1543 3 ATR[2, ATR$W_SIZE] = ATR$S_FPRO;
: 441 1544 3 ATR[2, ATR$L_ADDR] = FPRO;
: 442 1545 3 ATR[3, ATR$W_TYPE] = ATR$C_BAKDATE;    ! File backup date
: 443 1546 3 ATR[3, ATR$W_SIZE] = ATR$S_BAKDATE;
: 444 1547 3 ATR[3, ATR$L_ADDR] = BAK;
: 445 1548 3 ATR[4, ATR$W_TYPE] = ATR$C_UIC;      ! File owner UIC
: 446 1549 3 ATR[4, ATR$W_SIZE] = ATR$S_UIC;
: 447 1550 3 ATR[4, ATR$L_ADDR] = UIC;
: 448 1551 3 ATR[5, ATR$W_TYPE] = ATR$C_HEADER;    ! File header
: 449 1552 3 ATR[5, ATR$W_SIZE] = ATR$S_HEADER;
: 450 1553 3 ATR[5, ATR$L_ADDR] = HEADER;
: 451 1554 3 ATR[6, 0,0,32,0] = 0;                ! End of list
: 452 1555
: 453 1556 3 !+
: 454 1557 3 ! Back out to the previous directory level if the last lookup failed,
: 455 1558 3 ! to do the read attributes on the last directory file.
: 456 1559 3 !-
: 457 1560 3
: 458 1561 3 IF NOT .FINAL_STATUS
: 459 1562 3 THEN
: 460 1563 4 BEGIN
: 461 1564 4 FIB[FIB$W_FID_NUM] = .FIB[FIB$W_DID_NUM];
: 462 1565 4 FIB[FIB$W_FID_SEQ] = .FIB[FIB$W_DID_SEQ];
: 463 1566 4 FIB[FIB$W_FID_RVN] = .FIB[FIB$W_DID_RVN];
: 464 1567 3 END;
: 465 1568 3 FIB[FIB$W_DID_NUM] = 0;
: 466 1569 3 FIB[FIB$W_DID_SEQ] = 0;
: 467 1570 3 FIB[FIB$W_DID_RVN] = 0;
: 468 1571 3
: 469 1572 3 !+
: 470 1573 3 ! Read the attributes of the last directory file found so that they
: 471 1574 3 ! may be propagated to the directories created. If this fails with a
: 472 1575 3 ! NOSUCHFILE error, back out one level in case we had a dangling directory
: 473 1576 3 ! entry.
: 474 1577 3 !-
: 475 1578 3
: 476 1579 3 WHILE 1 DO
: 477 1580 4 BEGIN
: 478 1581 4 STATUS_3 = $QIOW(
: 479 1582 4 FUNC=IOS_ACCESS,
: 480 1583 4 CHAN=.OUTPUT_CHAN,
: 481 1584 4 IOSB=IOSB,
: 482 1585 4 P1=FIB_DESC,
: 483 1586 4 P5=ATR);
: 484 1587 4 IF .STATUS_3 THEN STATUS_3 = .IOSB[0];
: 485 1588 4 IF .STATUS_3
: 486 1589 4 THEN EXITLOOP
: 487 1590 4
: 488 1591 4 ! If we got a NOSUCHFILE error and a previous level exists,
: 489 1592 4 ! attempt to back up.
: 490 1593 4 !
: 491 1594 4 ELSE IF .STATUS_3 EQL SS$_NOSUCHFILE AND .PREV_ADDRESS NEQ 0
: 492 1595 4 THEN
```



```

: 493      1596 5      BEGIN
: 494      1597 5      FINAL STATUS = .STATUS_3;
: 495      1598 5      FIB[FIB$W_FID_NUM] = .FILE_ID[FID$W_NUM];
: 496      1599 5      FIB[FIB$W_FID_SEQ] = .FILE_ID[FID$W_SEQ];
: 497      1600 5      FIB[FIB$W_FID_RVN] = .FILE_ID[FID$W_RVN];
: 498      1601 5      DIR_LENGTH = .DIR_LENGTH - .PREV_ADDRESS + .DIR_ADDRESS;
: 499      1602 5      DIR_ADDRESS = .PREV_ADDRESS;
: 500      1603 5      PREV_ADDRESS = 0;
: 501      1604 5
: 502      1605 5      !+
: 503      1606 5      ! Locate the next directory name.
: 504      1607 5      !-
: 505      1608 5
: 506      1609 6      BEGIN ! block to use output registers
: 507      1610 6      REGISTER
: 508      1611 6          R0 = 0;
: 509      1612 6          R1 = 1;
: 510      1613 6      LOCAL
: 511      1614 6          NAME_LENGTH,
: 512      1615 6          NAME_ADDRESS;
: 513      1616 6
: 514      1617 6      NAME_ADDRESS = .DIR_ADDRESS;          ! Save starting point
: 515      1618 6      LOCCT%REF(%C'.'), DIR_LENGTH, .DIR_ADDRESS; R0, R1);
: 516      1619 6      NAME_LENGTH = .DIR_LENGTH - .R0;          ! Length preceding dot or end
: 517      1620 6      DIR_ADDRESS = .R1 + 1;          ! Prune to string following dot
: 518      1621 6      R0 = .R0 - 1;
: 519      1622 6      DIR_LENGTH = .R0;
: 520      1623 6
: 521      1624 6      !+
: 522      1625 6      ! Construct the directory name concatenated with '.DIR;1' in the name
: 523      1626 6      ! buffer, and a descriptor for this name in the name descriptor.
: 524      1627 6      !-
: 525      1628 6
: 526      1629 6      NAME_DESC[DSC$W_LENGTH] = .NAME_LENGTH + 6;
: 527      1630 6      CH$MOVE(6, DIR_TYP_VER, CH$MOVE7.NAME_LENGTH, .NAME_ADDRESS, NAME_BUFFER));
: 528      1631 5      END; ! block to use output registers
: 529      1632 5      END
: 530      1633 5
: 531      1634 5      ! Otherwise we can't back out. Report the error.
: 532      1635 5      !-
: 533      1636 4      ELSE
: 534      1637 5          BEGIN
: 535      1638 5              FINAL STATUS = .STATUS_3;
: 536      1639 5              LEAVE PROCESS;
: 537      1640 4          END;
: 538      1641 3      END;
: 539      1642 3
: 540      1643 3      !+
: 541      1644 3      ! Exit now in the /INCREMENTAL restore case where all directories exist.
: 542      1645 3      !-
: 543      1646 3
: 544      1647 3      IF .FINAL STATUS
: 545      1648 3      THEN LEAVE PROCESS;
: 546      1649 3
: 547      1650 3      !+
: 548      1651 3      ! Delete the file header attribute from the attribute list, since it is not
: 549      1652 3      ! valid (or necessary) for creates.
```



```

550 1653 3 !-
551 1654 3
552 1655 3 ATR[5, 0, 0, 32, 0] = 0;
553 1656 3 BAK[0] = BAK[1] = 0;
554 1657 4 IF (.QUAL[QUAL_REC0] OR .QUAL[QUAL_INCR]) ! If /RECORD or /INCREMENTAL
555 1658 3 AND .HEADER[FH2$B_STRUCLEV] EQL 2
556 1659 3 THEN
557 1660 4 BEGIN
558 1661 4 BAK[0] = .COM_O_BSRDATE[0];
559 1662 4 BAK[1] = .COM_O_BSRDATE[1];
560 1663 4 END;
561 1664 3
562 1665 3 !+
563 1666 3 Copy the file ID to FIB$W_DID to create the directory.
564 1667 3 !-
565 1668 3
566 1669 3 FIB[FIB$W_DID_NUM] = .FIB[FIB$W_FID_NUM];
567 1670 3 FIB[FIB$W_DID_SEQ] = .FIB[FIB$W_FID_SEQ];
568 1671 3 FIB[FIB$W_DID_RVN] = .FIB[FIB$W_FID_RVN];
569 1672 3
570 1673 3 !+
571 1674 3 Establish the allocation of the created directories. A Structure Level 1
572 1675 3 directory is allocated zero blocks; a Structure Level 2 directory is
573 1676 3 allocated one block. (This block is later initialized.) In both cases,
574 1677 3 the file is marked contiguous.
575 1678 3 !-
576 1679 3
577 1680 3 FIB[FIB$W_EXCTL] = FIB$M_EXTEND OR FIB$M_FILCON OR FIB$M_ALCON;
578 1681 3 FIB[FIB$W_NMCTL] = FIB$M_SUPERSEDE; ! To handle superseding -1 file ID
579 1682 3 IF .HEADER[FH2$B_STRUCLEV] EQL 2 THEN FIB[FIB$L_EXSZ] = 1;
580 1683 3
581 1684 3 !+
582 1685 3 Set up the end of file pointer. It points to the highest allocated block
583 1686 3 plus one (with a first free byte of zero). Note that EFBLK is stored in
584 1687 3 inverted format.
585 1688 3 !-
586 1689 3
587 1690 3 RECATTR[FAT$L_EFBLK] = ROT(.FIB[FIB$L_EXSZ] + 1, 16);
588 1691 3
589 1692 3 !+
590 1693 3 Establish the owner UIC of the created directories.
591 1694 3 !-
592 1695 3
593 1696 3 IF .OWNER_UIC NEQ 0
594 1697 3 THEN
595 1698 3 UIC = .OWNER_UIC;
596 1699 3
597 1700 3 !+
598 1701 3 Establish the file protection of the created directories.
599 1702 3 !-
600 1703 3
601 1704 3 FPRO = .FPRO OR %X'8888';
602 1705 3 IF .PROT_VALUE<0,16> NEQ %X'FFFF'
603 1706 3 THEN
604 1707 3 FPRO = .PROT_VALUE<0,16>;
605 1708 3
606 1709 3 !+

```



```

607 1710 3 ! Establish the default version limit of the created directories.
608 1711 3 !-
609 1712 3
610 1713 3 RECATTR[FAT$W_VERSIONS] = .MAX_VERSIONS;
611 1714 3
612 1715 3 !+
613 1716 3 ! Loop to create directories.
614 1717 3 !-
615 1718 3
616 1719 3 WHILE 1 DO
617 1720 4 BEGIN
618 1721 4
619 1722 4 !+
620 1723 4 ! Create and access the file.
621 1724 4 !-
622 1725 4
623 P 1726 4 FINAL STATUS = $QIOW(
624 P 1727 4 FUNC=IOS_CREATE OR IOSM_CREATE OR IOSM_ACCESS,
625 P 1728 4 CHAN=.OUTPUT_CHAN,
626 P 1729 4 IOSB=IOSB,
627 P 1730 4 P1=FIB_DESC,
628 P 1731 4 P2=NAME_DESC,
629 1732 4 P5=ATR);
630 1733 4 IF .FINAL_STATUS THEN FINAL STATUS = .IOSB[0];
631 1734 4 IF NOT .FINAL_STATUS THEN LEAVE PROCESS;
632 1735 4
633 1736 4 !+
634 1737 4 ! If the directory is Structure Level 2, write the first block of the
635 1738 4 ! file.
636 1739 4 !-
637 1740 4
638 1741 4 IF .HEADER[FH2$B_STRUCLEV] EQL 2
639 1742 4 THEN
640 1743 5 BEGIN
641 1744 5 LOCAL
642 1745 5 BLOCK_BUFFER: VECTOR[256,WORD]; ! Block buffer
643 1746 5
644 1747 5 BLOCK_BUFFER[0] = -1; ! End of block marker
645 1748 5 CH$FILL(0, 510, BLOCK_BUFFER[1]); ! Fill rest of block
646 P 1749 5 FINAL STATUS = $QIOW(
647 P 1750 5 FUNC=IOS_WRITEVBLK,
648 P 1751 5 CHAN=.OUTPUT_CHAN,
649 P 1752 5 IOSB=IOSB,
650 P 1753 5 P1=BLOCK_BUFFER,
651 P 1754 5 P2=512,
652 1755 5 P3=1);
653 1756 5 IF .FINAL_STATUS THEN FINAL STATUS = .IOSB[0];
654 1757 5 IF NOT .FINAL_STATUS THEN LEAVE PROCESS;
655 1758 4 END;
656 1759 4
657 1760 4 !+
658 1761 4 ! Deaccess the file.
659 1762 4 !-
660 1763 4
661 P 1764 4 FINAL STATUS = $QIOW(
662 P 1765 4 FUNC=IOS_DEACCESS,
663 P 1766 4 CHAN=.OUTPUT_CHAN,
```



```

664 1767 4      IOSB=IOSB);
665 1768 4      IF .FINAL_STATUS THEN FINAL_STATUS = .IOSB[0];
666 1769 4      IF NOT .FINAL_STATUS THEN LEAVE PROCESS;
667 1770 4
668 1771 4      !+
669 1772 4      ! If no more directory levels were specified, they have all been created.
670 1773 4      !-
671 1774 4
672 1775 4      IF .DIR_LENGTH LEQ 0
673 1776 4      THEN
674 1777 5          BEGIN
675 1778 5              FINAL_STATUS = $$$_CREATED;
676 1779 5              LEAVE PROCESS;
677 1780 4          END;
678 1781 4
679 1782 4      !+
680 1783 4      ! Locate the next directory name.
681 1784 4      !-
682 1785 4
683 1786 5      BEGIN ! block to use output registers
684 1787 5      REGISTER
685 1788 5          R0 = 0;
686 1789 5          R1 = 1;
687 1790 5      LOCAL
688 1791 5          NAME_LENGTH,
689 1792 5          NAME_ADDRESS;
690 1793 5
691 1794 5      NAME_ADDRESS = .DIR_ADDRESS;      ! Save starting point
692 1795 5      LOCCT%REF(%C'.'), DIR_LENGTH, .DIR_ADDRESS; R0, R1);
693 1796 5      NAME_LENGTH = .DIR_LENGTH - .R0;      ! Length preceding dot or end
694 1797 5      DIR_ADDRESS = .R1 + 1;      ! Prune to string following dot
695 1798 5      R0 = .R0 - 1;
696 1799 5      DIR_LENGTH = .R0;
697 1800 5
698 1801 5      !+
699 1802 5      ! Construct the directory name concatenated with '.DIR;1' in the name
700 1803 5      ! buffer, and a descriptor for this name in the name descriptor.
701 1804 5      !-
702 1805 5
703 1806 5      NAME_DESC[DSC$W_LENGTH] = .NAME_LENGTH + 6;
704 1807 5      CH$MOVE(6, DIR_TYP_VER, CH$MOVE%NAME_LENGTH, .NAME_ADDRESS, NAME_BUFFER));
705 1808 4      END; ! block to use output registers
706 1809 4
707 1810 4      !+
708 1811 4      ! Copy the file ID of the created directory to FIB$W_DID so that the next
709 1812 4      ! directory is cataloged in the directory just created.
710 1813 4      !-
711 1814 4
712 1815 4      FIB[FIB$W_DID_NUM] = .FIB[FIB$W_FID_NUM];
713 1816 4      FIB[FIB$W_DID_SEQ] = .FIB[FIB$W_FID_SEQ];
714 1817 4      FIB[FIB$W_DID_RVN] = .FIB[FIB$W_FID_RVN];
715 1818 4      END;
716 1819 3
717 1820 3      !+
718 1821 3      ! End of block that is exited when processing is complete. FINAL_STATUS
719 1822 3      ! contains the status that is to be returned to caller.
720 1823 3      !-
```


CREATEDIR
V04-000

Create directory
CREATE_DIR - create a directory file

K 1
15-Sep-1984 23:52:12
14-Sep-1984 11:53:50

VAX-11 Bliss-32 V4.0-742
[BACKUP.SRC]CREATEDIR.B32;1

Page 15
(3)

```

: 721      1824 3
: 722      1825 2 END; ! of block PROCESS
: 723      1826 2
: 724      1827 2 !+
: 725      1828 2 !- Return the status.
: 726      1829 2 !-
: 727      1830 2
: 728      1831 2 FILE_ID[FID$W_NUM] = .FIB[FIB$W_FID_NUM];
: 729      1832 2 FILE_ID[FID$W_SEQ] = .FIB[FIB$W_FID_SEQ];
: 730      1833 2 FILE_ID[FID$W_RVN] = .FIB[FIB$W_FID_RVN];
: 731      1834 2 BAKDATE[0] = .BAK[0];
: 732      1835 2 BAKDATE[1] = .BAK[1];
: 733      1836 2 RETURN .FINAL_STATUS;
: 734      1837 1 END;
```

.TITLE CREATEDIR Create directory
.IDENT \V04-000\

.PSECT COMMON,NOEXE, OVR,2

```
00000 GLOBAL_BASE:
      .BLKB 0
00000 FREE_LIST:
      .BLKB 8
00008 INPUT_WAIT:
      .BLKB 8
00010 REREAD_WAIT:
      .BLKB 8
00018 OUTPUT_WAIT:
      .BLKB 8
00020 JPI_UIC: .BLKB 4
00024 JPI_USERNAME:
      .BLKB 12
00030 JPI_DATE:
      .BLKB 8
00038 JPI_NODE_DESC:
      .BLKB 8
00040 JPI_CURPRIV:
      .BLKB 8
00048 SYI_VERSION:
      .BLKB 4
0004C SYI_SID: .BLKB 4
00050 RWSV_HOLD_LIST:
      .BLKB 8
00058 RWSV_CRC16:
      .BLKB 64
00098 RWSV_AUTODIN:
      .BLKB 64
000D8 RWSV_FILESET_ID:
      .BLKB 8
000E0 RWSV_VOLUME_ID:
      .BLKB 12
000EC RWSV_VOL_NUMBER:
      .BLKB 2
000EE RWSV_SEG_NUMBER:
      .BLKB 2
```


CREATEDIR
V04-000

Create directory
CREATE_DIR - create a directory file

L 1
15-Sep-1984 23:52:12
14-Sep-1984 11:53:50

VAX-11 Bliss-32 V4.0-742
[BACKUP.SRC]CREATEDIR.B32;1

Page 16
(3)

000.0 RWSV_FILE_NUMBER:
 .BLKB 4
000F4 RWSV_SAVE_QUAL:
 .BLKB 4
000F8 RWSV_SAVE_FAB:
 .BLKB 4
000FC RWSV_CHAN:
 .BLKB 4
00100 RWSV_XOR_BCB:
 .BLKB 4
00104 RWSV_IN_SEQ:
 .BLKB 4
00108 RWSV_IN_SEQ_0:
 .BLRB 4
0010C RWSV_IN_XOR_SEQ:
 .BLRB 4
00110 RWSV_IN_XOR_RFA:
 .BLRB 6
00116 RWSV_LOOKAHEAD:
 .BLKB 1
00117 RWSV_XOR_SIZE:
 .BLKB 1
00118 RWSV_IN_GROUP_SIZE:
 .BLKB 4
0011C RWSV_IN_ERRORS:
 .BLKB 2
0011E RWSV_IN_XORUSE:
 .BLKB 2
00120 RWSV_IN_ORGERR:
 .BLKB 8
00128 RWSV_IN_VBN:
 .BLKB 4
0012C RWSV_IN_VBN_0:
 .BLRB 4
00130 RWSV_ALLOC:
 .BLKB 4
00134 RWSV_EOF:
 .BLKB 4
00138 RWSV_OUT_SEQ:
 .BLKB 4
0013C RWSV_OUT_VBN:
 .BLKB 4
00140 RWSV_OUT_BLOCK_COUNT:
 .BLKB 4
00144 RWSV_OUT_ERRORS:
 .BLKB 2
00146 RWSV_SEQ_ERRORS:
 .BLKB 2
00148 RWSV_OUT_GROUP_COUNT:
 .BLKB 1
00149 RWSV_PADDING:
 .BLKB 3
0014C QUAL: .BLKB 112
001BC COM_SSNAME:
 .BLKB 8
001C4 COM_VALID_TYPES:
 .BLKB 2

CREATEDIR
V04-000

Create directory
CREATE_DIR - create a directory file

M 1
15-Sep-1984 23:52:12
14-Sep-1984 11:53:50

VAX-11 Bliss-32 V4.0-742
[BACKUP.SRC]CREATEDIR.B32;1

Page 17
(3)

001C6 COM_FLAGS: .BLKB 2
001C8 COM_PADDING: .BLKB 1
001C9 COM_BUFF_COUNT: .BLKB 1
001CA COM_I_SETCOUNT: .BLKB 1
001CB COM_O_SETCOUNT: .BLKB 1
001CC COM_I_STRUCNAME: .BLKB 12
001DB COM_O_STRUCNAME: .BLKB 12
001E4 COM_O_BSRDATE: .BLKB 8
001EC ALT_SSNAME: .BLKB 32
0020C INPUT_FUNC: .BLKB 1
0020D INPUT_RTYPE: .BLKB 1
0020E OUTPUT_FUNC: .BLKB 1
0020F FAST_STRUCLEV: .BLKB 1
00210 INPUT_BEG: .BLKB 0
00210 INPUT_CHAN: .BLKB 4
00214 INPUT_FLAGS: .BLKB 2
00216 INPUT_PADDING: .BLKB 2
00218 INPUT_FAB: .BLKB 4
0021C INPUT_NAM: .BLKB 4
00220 INPUT_BCB: .BLKB 4
00224 INPUT_QUAL: .BLKB 4
00228 INPUT_BAD: .BLKB 4
0022C INPUT_BLOCK: .BLKB 4
00230 INPUT_MAXBLOCK: .BLKB 4
00234 INPUT_MEDIA_ID: .BLKB 4
00238 INPUT_NAMEDESC: .BLKB 8
00240 INPUT_STATBLK: .BLKB 8
00248 INPUT_HDR_BEG: .BLKB 0
00248 INPUT_CREDATE:

CREATEDIR
V04-000

Create directory
CREATE_DIR - create a directory file

N 1
15-Sep-1984 23:52:12
14-Sep-1984 11:53:50

VAX-11 Bliss-32 V4.0-742
[BACKUP.SRC]CREATEDIR.B32;1

Page 18
(3)

00250	INPUT_REVDATE:	.BLKB	8
00258	INPUT_EXPDATE:	.BLKB	8
00260	INPUT_BAKDATE:	.BLKB	8
00268	INPUT_FILEOWNER:	.BLKB	8
0026C	INPUT_FILECHAR:	.BLKB	4
00270	INPUT_RECATTR:	.BLKB	4
00290	INPUT_HDR_END:	.BLKB	32
00290	INPUT_END:	.BLKB	0
00290	INPUT_PROC_LIST:	.BLKB	0
00294	INPUT_PLACEMENT:	.BLKB	4
0029C	INPUT_VBN_LIST:	.BLKB	8
002A4	INPUT_PLACE_LEN:	.BLKB	8
002A6	INPUT_PADDING_2:	.BLKB	2
002A8	OUTPUT_BEG:	.BLKB	2
002A8	OUTPUT_CHAN:	.BLKB	0
002AC	OUTPUT_FLAGS:	.BLKB	4
002AE	OUTPUT_PADDING:	.BLKB	2
002B0	OUTPUT_FAB:	.BLKB	2
002B4	OUTPUT_NAM:	.BLKB	4
002B8	OUTPUT_BCB:	.BLKB	4
002BC	OUTPUT_QUAL:	.BLKB	4
002C0	OUTPUT_BAD:	.BLKB	4
002C4	OUTPUT_BLOCK:	.BLKB	4
002C8	OUTPUT_MAXBLOCK:	.BLKB	4
002CC	OUTPUT_DEVGEO:	.BLKB	8
002D4	OUTPUT_ATTBUF:	.BLKB	144
00364	OUTPUT_END:	.BLKB	0
00364	LIST_TOTALFILES:	.BLKB	4

CREATEDIR
V04-000

Create directory
CREATE_DIR - create a directory file

B 2
15-Sep-1984 23:52:12
14-Sep-1984 11:53:50

VAX-11 Bliss-32 V4.0-742
[BACKUP.SRC]CREATEDIR.B32;1

Page 19
(3)

00368 LIST_TOTSIZE:
 .BLKB 4
0036C VERIFY_FAB:
 .BLKB 4
00370 VERIFY_USE_COUNT:
 .BLKB 4
00374 VERIFY_QUAL:
 .BLKB 4
00378 COMPARE_BCB:
 .BLKB 4
0037C FAST_BUFFER:
 .BLKB 4
00380 FAST_BUFFER_SIZE:
 .BLKB 4
00384 FAST_RVN:
 .BLKB 1
00385 FAST_PADDING:
 .BLKB 1
00386 DIR_VERLIMIT:
 .BLKB 2
00388 FAST_VOL_BEG:
 .BLKB 0
00388 FAST_IMAP_SIZE:
 .BLKB 4
0038C FAST_IMAP:
 .BLKB 4
00390 FAST_HDR_OFFSET:
 .BLKB 4
00394 FAST_BOOT_LBN:
 .BLKB 4
00398 FAST_VOL_END:
 .BLKB 0
00398 JOUR_BUFFER:
 .BLKB 4
0039C JOUR_DIR:
 .BLKB 4
003A0 JOUR_HIBLK:
 .BLKB 4
003A4 JOUR_EFBLK:
 .BLKB 4
003A8 JOUR_INBLK:
 .BLKB 4
003AC JOUR_FFBYTE:
 .BLKB 2
003AE JOUR_INBYTE:
 .BLKB 2
003B0 JOUR_STRUCT_LEV:
 .BLKB 2
003B2 JOUR_COUNT:
 .BLKB 1
003B3 JOUR_REVERSE:
 .BLKB 1
003B4 JOUR_EXSZ:
 .BLKB 2
003B6 JOUR_PADDING:
 .BLKB 2
003B8 CHKPT_HIGH_SP:

CREATEDIR
V04-000

Create directory
CREATE_DIR - create a directory file

C 2
15-Sep-1984 23:52:12
14-Sep-1984 11:53:50

VAX-11 Bliss-32 V4.0-742
[BACKUP.SRC]CREATEDIR.B32;1

Page 20
(3)

003BC	CHKPT_LOW_SP:	.BLKB	4
003C0	CHKPT_STACK:	.BLKB	4
003C4	CHKPT_VARS:	.BLKB	4
003C8	CHKPT_STATUS:	.BLKB	4
003CC	DIR_BEG:	.BLKB	0
003CC	DIR_CHAN:	.BLKB	4
003D0	DIR_NAM:	.BLKB	4
003D4	DIR_DEV_DESC:	.BLKB	4
003D8	DIR_SEL_DIR:	.BLKB	8
003E0	DIR_SEL_NTV:	.BLKB	8
003E8	DIR_STRUCLEV:	.BLKB	1
003E9	DIR_LEVELS:	.BLKB	1
003EA	DIR_FLAGS:	.BLKB	1
003EB	DIR_STATUS:	.BLKB	1
003EC	DIR_STRING:	.BLKB	320
0052C	DIR_STACK:	.BLKB	612
00790	DIR_SP:	.BLKB	4
00794	DIR_SEL_LATEST:	.BLKB	4
00798	DIR_END:	.BLKB	0
00798	DIR_SCANLIMIT:	.BLKB	36
007BC	INPUT_MTL:	.BLKB	4
007C0	OUTPUT_MTL:	.BLKB	4
007C4	CURRENT_MTL:	.BLKB	4
007C8	CURRENT_VCB:	.BLKB	4
007CC	CURRENT_WCB:	.BLKB	4
007D0	ACL_FIB_DESCR:	.BLKB	8
007D8	ACL_FIB:	.BLKB	64
00818	ACL_LENGTH:	.BLKB	4
0081C	ACL_BUFFER:	.BLKB	4
00820	CRYP_IN_CONTEXT:	.BLKB	4
00824	CRYP_OU_CONTEXT:	.BLKB	4

.BLKB 4
00828 CRYP_DA_CONTEXT:
.BLKB 4
0082C CRYP_DATA_ENCIV:
.BLKB 8
00834 CRYP_DATA_CODE:
.BLKB 4
00838 CRYP_DATA_KEY:
.BLKB 8
00840 CRYP_DATA_IV:
.BLKB 8
00848 CRYP_DATA_CKSM:
.BLKB 4

.PSECT CODE,NOWRT,2

31 3B 52 49 44 2E 00000 P.AAA: .ASCII \.DIR;1\ ;

DIR_TYP_VER= P.AAA
.EXTRN LIB\$CVT_OTB, LIB\$ INVFILSPE
.EXTRN SYSSPARSE, SYSSQIOW

OFFC 00000

0050	8F	00	5E	F954	CE	9E	00002	MOVAB	CREATE DIR, Save R2,R3,R4,R5,R6,R7,R8,R9,-	1198
			6E		00	2C	00007	MOVAB	R10,R11	
				B0	AD		0000E	MOVAB	-1708(SP), SP	
			B0	AD	8F	B0	00010	MOVAB	#0, (SP), #0, #80, \$RMS_PTR	1319
			C6	AD	02	90	00016	MOVW	#20483, \$RMS_PTR	
			CF	AD	02	90	0001A	MOVB	#2, \$RMS_PTR+22	
			D8	AD	02	90	0001E	MOVB	#2, \$RMS_PTR+31	
				50	CD	9E	0001E	MOVAB	NAM, \$RMS_PTR+40	
			DC	AD	04	AC	00024	MOVL	DEV DIR SPEC, R0	
			E4	AD	04	A0	00028	MOVL	4(R0), \$RMS_PTR+44	
0060	8F	00			60	90	0002D	MOVB	(R0), \$RMS_PTR+52	
					00	2C	00031	MOVAB	#0, (SP), #0, #96, \$RMS_PTR	1323
				FF50	CD		00038			
			FF50	CD	8F	B0	0003B	MOVW	#24578, \$RMS_PTR	
			FF58	CD	10	90	00042	MOVB	#16, \$RMS_PTR+8	
			FF5A	CD	01	8E	00047	MNEGB	#1, \$RMS_PTR+10	
			FF5C	CD	01	8E	0004C	MOVAB	ESA_BUFFER, \$RMS_PTR+12	
				FE50	CD	9E	0004C	MOVAB	ESA_BUFFER, \$RMS_PTR+12	
				B0	AD	9F	00053	PUSHAB	FAB	1331
		00000000G	00		01	FB	00056	CALLS	#1, SYSSPARSE	
			0A		50	E8	0005D	BLBS	STATUS_2, 1\$	1332
		0001C04A	8F		50	D1	00060	CMPL	STATUS_2, #114762	
					01	13	00067	BEQL	1\$	
						04	00069	RET		
			56	8A	AD	9A	0006A	MOVZBL	NAM+58, DIR_LENGTH	1336
			56		02	C2	0006E	SUBL2	#2, DIR_LENGTH	
	5A	98	AD		01	C1	00071	ADDL3	#1, NAM+72, DIR_ADDRESS	1337
	18	85	AD		05	E1	00076	BBC	#5, NAM+53, 2\$	1342
	50	98	BD		02	81	0007B	ADDB3	#2, @NAM+72, TERMINATOR	1353
	6A		56		50	3A	00080	LOCC	TERMINATOR, DIR_LENGTH, (DIR_ADDRESS)	1355
					37	13	00084	BEQL	4\$	
			57		51	D0	00086	MOVL	R1, REALDIR_ADDRESS	1359
	67	02	A7		50	28	00089	MOVAB	R0, 2(REALDIR_ADDRESS), (REALDIR_ADDRESS)	1369
			56		02	C2	0008E	SUBL2	#2, DIR_LENGTH	1373
					03	11	00091	BRB	3\$	

CREATEDIR
V04-000

Create directory
CREATE_DIR - create a directory file

E 2
15-Sep-1984 23:52:12
14-Sep-1984 11:53:50

VAX-11 Bliss-32 V4.0-742
[BACKUP.SRC]CREATEDIR.B32;1

Page 22
(3)

2A	86	57	AD	4080	5A	D0	00093	2\$:	MOVL	DIR_ADDRESS, REALDIR_ADDRESS	1380
					03	E1	00096	3\$:	BBC	#3, NAM+54, 5\$	1384
					8F	BB	0009B		PUSHR	#^M<R7,SP>	1390
					03	DD	0009F		PUSHL	#3	
	00000000G	00			03	FB	000A1		CALLS	#3, LIB\$CVT_OTB	
		12			50	E9	000A8		BLBC	R0, 4\$	
				04	AE	9F	000AB		PUSHAB	MEMBER	1396
				03	A7	9F	000AE		PUSHAB	3(REALDIR_ADDRESS)	
					03	DD	000B1		PUSHL	#3	
	00000000G	00			03	FB	000B3		CALLS	#3, LIB\$CVT_OTB	
		08			50	E8	000BA		BLBS	R0, 5\$	
		50	00000000G		8F	D0	000BD	4\$:	MOVL	#LIB\$_INVFILSPE, R0	1398
					04	000C4			RET		
0040	8F	00	6E		00	2C	000C5	5\$:	MOVC5	#0, (SP), #0, #64, FIB	1406
				0250	CE		000CC				
	0250	CE		0501	8F	3C	000CF		MOVZWL	#1281, FIB	1407
	0264	CE		0400	8F	B0	000D6		MOVW	#1024, FIB+20	1408
	0254	CE	00040004		8F	D0	000DD		MOVL	#262148, FIB+4	1409
	0248	CE		40	8F	9A	000E6		MOVZBL	#64, FIB_DESC	1416
	024C	CE		0250	CE	9E	000EC		MOVAB	FIB, FIB_DESC+4	1417
	FE1C	CD	FE20		CD	9E	000F3		MOVAB	NAME_BUFFER, NAME_DESC+4	1423
					5B	D4	000FA		CLRL	PREV_ADDRESS	1436
		59	0254		CE	3C	000FC		MOVZWL	FIB+4, R9	1445
		57	14		AC	D0	00101		MOVL	FILE_ID, R7	1503
	025A	CE			59	B0	00105	6\$:	MOVW	R9, FIB+10	1445
	025C	CE	0256		CE	D0	0010A		MOVL	FIB+6, FIB+12	1446
		53			5A	D0	00111		MOVL	DIR_ADDRESS, NAME_ADDRESS	1462
6A		56			2E	3A	00114		LOCC	#46, DIR_LENGTH, (DIR_ADDRESS)	1463
52		56			50	C3	00118		SUBL3	R0, DIR_LENGTH, NAME_LENGTH	1464
		5A	01		A1	9E	0011C		MOVAB	1(R1), DIR_ADDRESS	1465
		56			70	9E	00120		MOVAB	-(R0), DIR_LENGTH	1467
FE18	CD	52			06	A1	00123		ADDW3	#6, NAME_LENGTH, NAME_DESC	1474
FE20	CD	63			52	28	00129		MOVC3	NAME_LENGTH, (NAME_ADDRESS), NAME_BUFFER	1475
	63	FEC6	CF		06	28	0012F		MOVC3	#6, DIR_TYP_VER, (R3)	
					7E	7C	00135		CLRQ	-(SP)	1488
					7E	7C	00137		CLRQ	-(SP)	
				FE18	CD	9F	00139		PUSHAB	NAME_DESC	
				025C	CE	9F	0013D		PUSHAB	FIB_DESC	
					7E	7C	00141		CLRQ	-(SP)	
				0228	CE	9F	00143		PUSHAB	IOSB	
					32	DD	00147		PUSHL	#50	
				00000000'	EF	DD	00149		PUSHL	OUTPUT_CHAN	
					7E	D4	0014F		CLRL	-(SP)	
	00000000G	00			0C	FB	00151		CALLS	#12, SYS\$QIOW	
		58			50	D0	00158		MOVL	R0, FINAL_STATUS	1489
		05			58	E9	0015B		BLBC	FINAL_STATUS, 7\$	
		58	0208		CE	3C	0015E		MOVZWL	IOSB, FINAL_STATUS	1490
		59	0254		CE	3C	00163	7\$:	MOVZWL	FIB+4, R9	
	FFFF	8F			59	B1	00168		CMPW	R9, #65535	
					16	12	0016D		BNEQ	8\$	
	FFFF	8F	0256		CE	B1	0016F		CMPW	FIB+6, #65535	1491
					0D	12	00176		BNEQ	8\$	
	FF	8F	0259		CE	91	00178		CMPB	FIB+9, #255	1492
					05	12	0017E		BNEQ	8\$	
		58	0910		8F	3C	00180		MOVZWL	#2320, FINAL_STATUS	1493
	00000910	8F			58	D1	00185	8\$:	CMPL	FINAL_STATUS, #2320	1494
					28	13	0018C		BEQL	11\$	

		03	58	E8	0018E	BLBS	FINAL_STATUS, 9\$	1496
			0299	31	00191	BRW	28\$	
51		51	FE18	CD	3C 00194	MOVZWL	NAME_DESC, R1	1502
		5A		51	C3 00199	SUBL3	R1, DIR_ADDRESS, R1	
		5B	05	A1	9E 0019D	MOVAB	5(R1), PREV_ADDRESS	
		67	025A	CE	D0 001A1	MOVL	FIB+10, (R7)	1503
	04	A7	025E	CE	B0 001A6	MOVW	FIB+14, 4(R7)	1505
				56	D5 001AC	TSTL	DIR_LENGTH	1517
				03	15 001AE	BLEQ	10\$	
				FF52	31 001B0	BRW	6\$	
		58		01	D0 001B3	MOVL	#1, FINAL_STATUS	1520
0210		CE	00040020	8F	D0 001B6	MOVL	#262176, ATR	1537
0214		CE	FDF8	CD	9E 001BF	MOVAB	RECATTR, ATR+4	1538
0218		CE	00030004	8F	D0 001C6	MOVL	#196612, ATR+8	1540
021C		CE	FDF4	CD	9E 001CF	MOVAB	UCHAR, ATR+12	1541
0220		CE	00160002	8F	D0 001D6	MOVL	#1441794, ATR+16	1543
0224		CE	FDF0	CD	9E 001DF	MOVAB	FPRO, ATR+20	1544
0228		CE	00140008	8F	D0 001E6	MOVL	#1310728, ATR+24	1546
022C		CE	FDE4	CD	9E 001EF	MOVAB	BAK, ATR+28	1547
0230		CE	00150004	8F	D0 001F6	MOVL	#1376260, ATR+32	1549
0234		CE	FDEC	CD	9E 001FF	MOVAB	UIC, ATR+36	1550
0238		CE	000A0200	8F	D0 00206	MOVL	#655872, ATR+40	1552
023C		CE	0290	CE	9E 0020F	MOVAB	HEADER, ATR+44	1553
			0240	CE	D4 00216	CLRL	ATR+48	1554
		0E		58	E8 0021A	BLBS	FINAL_STATUS, 12\$	1561
0254		CE	025A	CE	D0 0021D	MOVL	FIB+10, FIB+4	1564
0258		CE	025E	CE	B0 00224	MOVW	FIB+14, FIB+8	1566
			025A	CE	D4 0022B	CLRL	FIB+10	1568
			025E	CE	B4 0022F	CLRW	FIB+14	1570
				7E	D4 00233	CLRL	-(SP)	1586
			0214	CE	9F 00235	PUSHAB	ATR	
				7E	7C 00239	CLRQ	-(SP)	
				7E	D4 0023B	CLRL	-(SP)	
			025C	CE	9F 0023D	PUSHAB	FIB_DESC	
				7E	7C 00241	CLRQ	-(SP)	
			0228	CE	9F 00243	PUSHAB	IOSB	
				32	DD 00247	PUSHL	#50	
			00000000'	EF	DD 00249	PUSHL	OUTPUT_CHAN	
				7E	D4 0024F	CLRL	-(SP)	
00000000G		00		0C	FB 00251	CALLS	#12, SYSSQIOW	
		59		50	D0 00258	MOVL	R0, STATUS_3	
		08		59	E9 0025B	BLBC	STATUS_3, T4\$	1587
		59	0208	CE	3C 0025E	MOVZWL	IOSB, STATUS_3	
		52		59	E8 00263	BLBS	STATUS_3, 17\$	1588
		58		59	D0 00266	MOVL	STATUS_3, FINAL_STATUS	1597
00000910		8F		59	D1 00269	CMPL	STATUS_3, #2320	1594
				03	13 00270	BEQL	16\$	
				01B8	31 00272	BRW	28\$	
				5B	D5 00275	TSTL	PREV_ADDRESS	
				F9	13 00277	BEQL	15\$	
		0254	CE	67	D0 00279	MOVL	(R7), FIB+4	1598
		0258	CE	A7	B0 0027E	MOVW	4(R7), FIB+8	1600
50				5B	C3 00284	SUBL3	PREV_ADDRESS, DIR_LENGTH, R0	1601
56				5A	C1 00288	ADDL3	DIR_ADDRESS, R0, DIR_LENGTH	
				5B	D0 0028C	MOVL	PREV_ADDRESS, DIR_ADDRESS	1602
				5B	D4 0028F	CLRL	PREV_ADDRESS	1603
				5A	D0 00291	MOVL	DIR_ADDRESS, NAME_ADDRESS	1617

CREATEDIR
V04-000

Create directory
CREATE_DIR - create a directory file

G 2
15-Sep-1984 23:52:12
14-Sep-1984 11:53:50

VAX-11 Bliss-32 V4.0-742
[BACKUP.SRC]CREATEDIR.B32;1

Page 24
(3)

6A	56	2E	3A	00294	LOCC	#46, DIR_LENGTH, (DIR_ADDRESS)	1618
52	56	50	C3	00298	SUBL3	R0, DIR_LENGTH, NAME_LENGTH	1619
	5A	A1	9E	0029C	MOVAB	1(R1), DIR_ADDRESS	1620
	56	70	9E	002A0	MOVAB	-(R0), DIR_LENGTH	1622
FE18	CD	06	A1	002A3	ADDW3	#6, NAME_LENGTH, NAME_DESC	1629
FE20	CD	52	28	002A9	MOVCS	NAME_LENGTH, (NAME_ADDRESS), NAME_BUFFER	1630
63	FD46	06	28	002AF	MOVCS	#6, DIR_TYP_VER, (R3)	1594
		FF7B	31	002B5	BRW	13\$	1647
	B7	58	E8	002B8	BLBS	FINAL STATUS, 15\$	1655
		CE	D4	002BB	CLRL	ATR+40	1656
	0238	CD	D4	002BF	CLRL	BAK+4	1657
	FDE8	CD	D4	002C3	CLRL	BAK	1658
	FDE4	CD	D4	002C7	BBS	#6, QUAL+12, 18\$	1661
08 00000000'	EF	06	E0	002C7	BBC	#4, QUAL+10, 19\$	1662
19 00000000'	EF	04	E1	002CF	CMPB	HEADER+7, #2	1669
	02	0297	CE	91	BNEQ	19\$	1671
			12	002DC	MOVL	COM_O_BSRDATE, BAK	1681
	FDE4	CD	00000000'	EF	DO	COM_O_BSRDATE+4, BAK+4	1682
	FDE8	CD	00000000'	EF	DO	FIB+4, FIB+10	1690
	025A	CE	0254	CE	DO	FIB+8, FIB+14	1696
	025E	CE	0258	CE	BO	#8717312, FIB+20	1698
	0264	CE	00850400	8F	DO	HEADER+7, #2	1704
		02	0297	CE	91	20\$	1705
				05	12	#1, FIB+24	1707
	0268	CE		01	DO	#1, FIB+24, R0	1713
FE00	50	CE		01	C1	#16, R0, RECATTR+8	1732
CD	0268	50		10	9C	OWNER_UIC	1733
			08	AC	D5	21\$	1734
				06	13	22\$	1741
	FDEC	CD	08	AC	DO	OWNER UIC, UIC	1747
	FDF0	CD	8888	8F	A8	#34952, FPRO	1748
	FFFF	8F	0C	AC	B1	PROT_VALUE, #65535	1755
				06	13	22\$	
	FDF0	CD	0C	AC	BO	PROT VALUE, FPRO	
	FE16	CD	10	AC	BO	MAX VERSIONS, RECATTR+30	
				7E	D4	-(SP)	
			0214	CE	9F	PUSHAB	
				7E	7C	CLRQ	
			FE18	CD	9F	PUSHAB	
			025C	CE	9F	PUSHAB	
				7E	7C	CLRQ	
			0228	CE	9F	PUSHAB	
			F3	8F	9A	MOVZBL	
			00000000'	EF	DD	PUSHL	
				7E	D4	CLRL	
				0C	FB	CALLS	
			00000000G	50	DO	#12, SYSSQIOW	
				58	E9	MOVL	
				76	3C	BLBC	
			0208	CE	3C	MOVZWL	
				58	E9	BLBC	
			0297	CE	91	CMPB	
				3F	12	BNEQ	
				01	AE	MNEGW	
01FE	8F		08	AE	2C	MOVCS	
				00	2C		
			0A	AE			
				7E	7C	CLRQ	
				01	7D	MOVQ	
			0200	8F	3C	MOVZWL	

CREATEDIR
V04-000

Create directory
CREATE_DIR - create a directory file

H 2
15-Sep-1984 23:52:12
14-Sep-1984 11:53:50

VAX-11 Bliss-32 V4.0-742
[BACKUP.SRC]CREATEDIR.B32;1

Page 25
(3)

		1C	AE	9F	0039A	PUSHAB	BLOCK_BUFFER	:	
			7E	7C	0039D	CLRQ	-(SP)	:	
		0228	CE	9F	0039F	PUSHAB	IOSB	:	
			30	DD	003A3	PUSHL	#48	:	
		00000000'	EF	DD	003A5	PUSHL	OUTPUT_CHAN	:	
			7E	D4	003AB	CLRL	-(SP)	:	
00000000G	00		0C	FB	003AD	CALLS	#12, SYSSQIOW	:	
	58		50	D0	003B4	MOVL	R0, FINAL_STATUS	:	
	73		58	E9	003B7	BLBC	FINAL_STATUS, 28\$	:	1756
	58	0208	CE	3C	003BA	MOVZWL	IOSB, FINAL_STATUS	:	
	6B		58	E9	003BF	BLBC	FINAL_STATUS, 28\$	:	1757
			7E	7C	003C2	CLRQ	-(SP)	:	1767
			7E	7C	003C4	CLRQ	-(SP)	:	
			7E	7C	003C6	CLRQ	-(SP)	:	
			7E	7C	003C8	CLRQ	-(SP)	:	
		0228	CE	9F	003CA	PUSHAB	IOSB	:	
			34	DD	003CE	PUSHL	#52	:	
		00000000'	EF	DD	003D0	PUSHL	OUTPUT_CHAN	:	
			7E	D4	003D6	CLRL	-(SP)	:	
00000000G	00		0C	FB	003D8	CALLS	#12, SYSSQIOW	:	
	58		50	D0	003DF	MOVL	R0, FINAL_STATUS	:	
	48		58	E9	003E2	BLBC	FINAL_STATUS, 28\$	:	1768
	58	0208	CE	3C	003E5	MOVZWL	IOSB, FINAL_STATUS	:	
	40		58	E9	003EA	BLBC	FINAL_STATUS, 28\$	:	1769
			56	D5	003ED	TSTL	DIR_LENGTH	:	1775
			07	14	003EF	BGTR	27\$	:	
	58	0619	8F	3C	003F1	MOVZWL	#1561, FINAL_STATUS	:	1778
			35	11	003F6	BRB	28\$	:	1779
	53		5A	D0	003F8	MOVL	DIR_ADDRESS, NAME_ADDRESS	:	1794
6A	56		2E	3A	003FB	LOCC	#46, DIR_LENGTH, (DIR_ADDRESS)	:	1795
52	56		50	C3	003FF	SUBL3	R0, DIR_LENGTH, NAME_LENGTH	:	1796
	5A	01	A1	9E	00403	MOVAB	1(R1), DIR_ADDRESS	:	1797
	56		70	9E	00407	MOVAB	-(R0), DIR_LENGTH	:	1799
FE18	CD		06	A1	0040A	ADDW3	#6, NAME_LENGTH, NAME_DESC	:	1806
FE20	CD		52	28	00410	MOVC3	NAME_LENGTH, (NAME_ADDRESS), NAME_BUFFER	:	1807
	63		06	28	00416	MOVC3	#6, DIR_TYP_VER, (R3)	:	
	FBD		CE	D0	0041C	MOVL	FIB+4, FIB+10	:	1815
	025A	0254	CE	B0	00423	MOVW	FIB+8, FIB+14	:	1817
	025E	0258	FF18	31	0042A	BRW	23\$	:	1719
			CE	D0	0042D	MOVL	FIB+4, (R7)	:	1831
	67	0254	CE	B0	00432	MOVW	FIB+8, 4(R7)	:	1833
	04	18	AC	D0	00438	MOVL	BAKDATE, R0	:	1834
	50	FDE4	CD	D0	0043C	MOVL	BAK, (R0)	:	
	60	FDE8	CD	D0	00441	MOVL	BAK+4, 4(R0)	:	1835
	04		58	D0	00447	MOVL	FINAL_STATUS, R0	:	1836
			04	0044A	RET			:	1837

; Routine Size: 1099 bytes, Routine Base: CODE + 0006

CREATEDIR
V04-000

Create directory
CREATE_DIR - create a directory file

1 2
15-Sep-1984 23:52:12
14-Sep-1984 11:53:50

VAX-11 Bliss-32 V4.0-742
[BACKUP.SRC]CREATEDIR.B32;1

Page 26
(4)

: 736 1838 1 END
: 737 1839 0 ELUDOM

PSECT SUMMARY

Name	Bytes	Attributes
COMMON	2124	NOVEC, WRT, RD, NOEXE, NOSHR, LCL, REL, OVR, NOPIC, ALIGN(2)
CODE	1105	NOVEC, NOWRT, RD, EXE, NOSHR, LCL, REL, CON, NOPIC, ALIGN(2)

Library Statistics

File	----- Total	Symbols Loaded	----- Percent	Pages Mapped	Processing Time
_\$255SDUA28:[SYSLIB]LIB.L32;1	18619	116	0	1000	00:02.1

COMMAND QUALIFIERS

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

: Size: 1099 code + 2130 data bytes
: Run Time: 00:34.1
: Elapsed Time: 01:45.6
: Lines/CPU Min: 3234
: Lexemes/CPU-Min: 38035
: Memory Used: 496 pages
: Compilation Complete

0010 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

0011 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

