

[illegible]

3

Sy

MT

MT
MT

MT

MT

MT
MTMT
MTMT
MTMT
MTMT
MTMT
MT

MT

MT

MT
MT

MI
MT

MT
MT

MT
MT

MT
MT

MT
MT

MT

111

227

MT

MT
MT

MT

MT

M1
M1MT
MT

M1

M1

10

1

10

100

10

1

```

000000  TTTTTTTTTT  SSSSSSSS  PPPPPPPP  000000  WW  WW  GGGGGGGG  JJ
000000  TTTTTTTTTT  SSSSSSSS  PPPPPPPP  000000  WW  WW  GGGGGGGG  JJ
00      00      SS      PP      PP  00      00  WW  WW  GG  JJ
00      00      SS      PP      PP  00      00  WW  WW  GG  JJ
00      00      SS      PP      PP  00      00  WW  WW  GG  JJ
00      00      SS      PP      PP  00      00  WW  WW  GG  JJ
00      00      SS      PP      PP  00      00  WW  WW  GG  JJ
00      00      SS      PP      PP  00      00  WW  WW  GG  JJ
00      00      SS      PP      PP  00      00  WW  WW  GG  JJ
00      00      SS      PP      PP  00      00  WW  WW  GG  JJ
00      00      SS      PP      PP  00      00  WWW  WWW  GG  JJ
00      00      SS      PP      PP  00      00  WWW  WWW  GG  JJ
000000  TT      SSSSSSSS  PP      000000  WW  WW  GGGGGG  JJJJJJ
000000  TT      SSSSSSSS  PP      000000  WW  WW  GGGGGG  JJJJJJ

```

```

LL      IIIIIII  SSSSSSSS
LL      IIIIIII  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  IIIIIII  SSSSSSSS
LLLLLLLLLLLL  IIIIIII  SSSSSSSS

```


(2) 51
(3) 65
(4) 108

HISTORY ; Detailed Current Edit History
DECLARATIONS
OTSS\$POWGJ - G REAL*8 ** INTEGER*4

```

0000 1      .TITLE  OTSSPOWGJ - G REAL*8 ** INTEGER*4 power routine
0000 2      .IDENT  /1-005/      ; File: OTSPOWGJ.MAR Edit: SBL1005
0000 3
0000 4
0000 5      *****
0000 6      *
0000 7      *  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
0000 8      *  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
0000 9      *  ALL RIGHTS RESERVED.
0000 10     *
0000 11     *  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
0000 12     *  ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
0000 13     *  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
0000 14     *  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
0000 15     *  OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
0000 16     *  TRANSFERRED.
0000 17     *
0000 18     *  THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
0000 19     *  AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
0000 20     *  CORPORATION.
0000 21     *
0000 22     *  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
0000 23     *  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
0000 24     *
0000 25     *
0000 26     *****
0000 27
0000 28
0000 29     FACILITY: Language support library - user callable
0000 30     ++
0000 31     ABSTRACT:
0000 32
0000 33         G REAL*8 base to INTEGER*4 power.
0000 34         Floating overflow and underflow can occur.
0000 35         Undefined exponentation can occur if base is 0 and power
0000 36         is 0 or negative.
0000 37
0000 38
0000 39     --
0000 40
0000 41     VERSION: 1
0000 42
0000 43     HISTORY:
0000 44     AUTHOR:
0000 45         Steven B. Lionel, 6-Feb-79: Version 1
0000 46
0000 47
0000 48
0000 49

```


OTSS\$POWGJ
1-005

I 16
- G REAL*8 ** INTEGER*4 power routine 16-SEP-1984 01:59:33 VAX/VMS Macro V04-00
HISTORY ; Detailed Current Edit History 6-SEP-1984 11:28:16 [MTHRTL.SRC]OTSS\$POWGJ.MAR;1 Page 2
(2)

```
0000 51 .SBTTL HISTORY ; Detailed Current Edit History
0000 52
0000 53
0000 54 : Edit History for Version 1 of OTSS$POWGJ
0000 55 : 1-002 SBL1002 - Declare externals. SBL 17-May-1979
0000 56 : 1-003 - Correct some comments. JBS 30-JUL-1979
0000 57 : 1-001 - Adapted from OTSS$POWDJ version 1-001. SBL 06-Feb-79
0000 58 : 1-004 - Add handlers to catch SSS$FLT0VF and SSS$FLTDIV, and signal
0000 59 : MTH$_FLOOVEMAT or MTH$_FLOUNDMAT instead, depending on the context.
0000 60 : Also disable IV and change BLBS/ASHL at EXPGTR to BBSC/ROTL for
0000 61 : uniformity with OTSS$POWRJ. JAW 26-Feb-1980.
0000 62 : 1-005 - Use general mode addressing. SBL 30-Nov-1981
0000 63 :
```

```

0000 65      .SBTTL  DECLARATIONS
0000 66
0000 67 :
0000 68 : INCLUDE FILES:
0000 69 :
0000 70 :
0000 71 :
0000 72 : EXTERNAL SYMBOLS:
0000 73 :
0000 74 :
0000 75      .DSABL  GBL
0000 76      .EXTRN  MTH$K_UNDEXP, MTH$K_FLOOVEMAT, MTH$K_FLOUNDMAT
0000 77      .EXTRN  MTH$$SIGNAL ; Math error routine
0000 78      .EXTRN  SSS_FLTOVF, SSS_FLTOVF_F, SSS_FLTDIV, SSS_FLTDIV_F, SSS_CONTINUE
0000 79 :
0000 80 : MACROS:
0000 81 :
0000 82      $CHFDEF ; Define condition handler symbols.
0000 83      $SFDEF  ; Define stack frame symbols.
0000 84      $PSLDEF ; Define program status longword
0000 85                ; symbols.
0000 86
0000 87 :
0000 88 : EQUATED SYMBOLS:
0000 89 :
00000004 0000 90      base = 4 ; base input formal - by-value
0000000C 0000 91      exp = 12 ; exponent intpu formal - by-value
0000 92                ; Note: G_floating by-value violates
0000 93                ; calling standard, but ok since this
0000 94                ; routine is a code support routine (OTSS)
0000 95
0000 96 :
0000 97 : OWN STORAGE:
0000 98 :
0000 99 :
0000 100 :
0000 101 : PSECT DECLARATIONS:
0000 102 :
0000 103 :
00000000 0000 104      .PSECT  _OTSS$CODE PIC,SHR,LONG,EXE,NOWRT
0000 105                ; program section for OTSS$ code
0000 106

```



```

0000 108      .SBTTL OTSS$POWGJ - G REAL*8 ** INTEGER*4
0000 109
0000 110      :++
0000 111      : FUNCTIONAL DESCRIPTION:
0000 112      :
0000 113      :   G_floating result = G_floating base ** signed longword exponent
0000 114      :   The G_floating result is given by:
0000 115      :
0000 116      :   base      exponent      result
0000 117      :
0000 118      :   any      > 0      product (base * 2**i) where i is each
0000 119      :                               non-zero bit position in exponent
0000 120      :
0000 121      :   > 0      = 0      1.0
0000 122      :   = 0      = 0      Undefined exponentiation
0000 123      :   < 0      = 0      1.0
0000 124      :
0000 125      :   > 0      < 0      1.0 / product (base * 2**i)
0000 126      :                               where i is each non-zero bit position
0000 127      :                               in |exponent|
0000 128      :   = 0      < 0      Undefined exponentiation
0000 129      :   < 0      < 0      1.0 / product (base * 2**i)
0000 130      :                               where i is each non-zero bit position
0000 131      :                               in |exponent|
0000 132      :
0000 133      :   Floating overflow can occur on either of the two MULG's. If this
0000 134      :   happens when the exponent is less than zero, the exception is caught by
0000 135      :   a local condition handler named EXC_HNDLR_UNDER, which sets the result
0000 136      :   to 0.0 and either signals MTH$ FLOUNDMAT (if FU is enabled in the
0000 137      :   caller's PSW) or continues at POWGJX. If it happens when the exponent
0000 138      :   is greater than zero, the exception is caught by a local condition
0000 139      :   handler named EXC_HNDLR_OVER, which sets the result to the reserved
0000 140      :   operand (-0.0) and signals MTH$_FLOOVEMAT.
0000 141      :
0000 142      :   Floating overflow and floating divide by zero can occur on the DIVG.
0000 143      :   These exceptions are caught by EXC_HNDLR_OVER, which sets the result to
0000 144      :   the reserved operand (-0.0) and signals MTH$_FLOOVEMAT.
0000 145      :
0000 146      :   Undefined exponentiation occurs if base is 0 and
0000 147      :   exponent is 0 or negative.
0000 148      :
0000 149      : CALLING SEQUENCE:
0000 150      :
0000 151      :   Power.wg.v = OTSS$POWGJ (base.rg.v, exponent.rl.v)
0000 152      :
0000 153      : INPUT PARAMETERS:
0000 154      :   base      - G REAL*8 base
0000 155      :   exponent  - INTEGER*4 exponent
0000 156      :
0000 157      : IMPLICIT INPUTS:
0000 158      :   The setting of FU in the caller's PSW.
0000 159      :
0000 160      : OUTPUT PARAMETERS:
0000 161      :   NONE
0000 162      :
0000 163      : IMPLICIT OUTPUTS:
0000 164      :   NONE

```

```
0000 165 :  
0000 166 : FUNCTION VALUE:  
0000 167 :  
0000 168 : G_floating base ** signed longword exponent  
0000 169 :  
0000 170 : SIDE EFFECTS:  
0000 171 :  
0000 172 : Signals MTH$ FLOOVEMAT if floating overflow occurs on either of the two  
0000 173 : MULG's when exponent > 0, or if floating overflow or divide by zero  
0000 174 : occurs on the DIVG.  
0000 175 : Signals MTH$ FLOUNDMAT if floating overflow occurs on either of the two  
0000 176 : MULG's when exponent < 0 and caller has FU enabled.  
0000 177 : SIGNALS MTH$ UNDEXP (82 = ' UNDEFINED EXPONENTATION') if  
0000 178 : base is 0 and exponent is 0 or negative.  
0000 179 :  
0000 180 :--  
0000 181 :  
0000 182 :  
0000 183 :  
001C 0000 184 .ENTRY OTSS$POWGJ, ^M<R2, R3, R4>  
0002 185 : Disable integer overflow. (Occurs on  
0002 186 : maximum negative exponent.)  
6D 6C'AF 9E 0002 187 MOVAB B^EXC_HNDLR_OVER, (FP) : Translate exceptions to  
0006 188 : MTH$ FLOOVEMAT.  
52 50 08 50FD 0006 189 MOVG #1, R0 : R0/R1 = initial result  
52 04 AC 50FD 000A 190 MOVG base(AP), R2 : R2/R3 = base  
54 0C AC D0 000F 191 MOVL exp(AP), R4 : R4 = exponent  
6D 5D'AF 9E 0013 192 BGTR EXPGTR : branch if exponent > 0  
0015 193 MOVAB B^EXC_HNDLR_UNDER, (FP) : Translate exceptions to  
0019 194 : MTH$ FLOUNDMAT.  
0019 195 :  
52 53FD 0019 196 TSTG R2 : test base  
2F 13 001C 197 BEQL UNDEFINED : undefined 0**0 or 0**(-n)  
54 54 CE 001E 198 MNEGL R4, R4 : R4 = !exponent!  
29 13 0021 199 BEQL POWGJX : if exponent is 0, return R0 = 1.0  
0023 200 :  
0023 201 :+  
0023 202 : Exponent is > 0 or (exponent is =< 0 and base is not = 0 -- use !exponent!)  
0023 203 :--  
0023 204 :  
54 0C 54 00 E4 0023 205 EXPGTR: BBSC #0, R4, PARTIAL : branch if !exponent! is odd  
54 54 FF 8F 9C 0027 206 SQUAR: ROTL #-1, R4, R4 : R4 = !exponent!/2  
52 52 44FD 002C 207 SQUAR1: MULG2 R2, R2 : R2/R3 = current power of base  
0030 208 : Floating overflow will trap or fault  
0030 209 : and signal SS$ FLTOVF or SS$ FLTOVF_F.  
F4 54 E9 0030 210 BLBC R4, SQUAR : branch if next bit in !exponent! is 0  
0033 211 :  
0033 212 :+  
0033 213 : Here when bit i of !exponent! is a 1.  
0033 214 : Partial result = partial result * (base * 2**i)  
0033 215 :--  
0033 216 :  
0033 217 PARTIAL:  
54 54 50 52 44FD 0033 218 MULG2 R2, R0 : R0/R1 = new partial result  
54 54 FF 8F 78 0037 219 ASHL #-1, R4, R4 : R4 = !exponent!/2  
EE 12 003C 220 BNEQ SQUAR1 : loopback if more exponent bits are 1  
003E 221
```



```

      OC AC D5 003E 222      TSTL      exp(AP)      ; test sign of exponent
      09 14 0041 223      BGTR      POWGJX      ; if exponent > 0, return R0
      6D 6C'AF 9E 0043 224      MOVAB      B^EXC_HNDLR_OVER, (FP) ; Translate exceptions to
      50 08 50 47FD 0047 225      DIVG3      R0, #1, R0      ; MTH$_FLOOVEMAT.
      04 004C 226      POWGJX: RET      ; R0/RT = 1.0/result
      004D 227      ; return, result in R0
      004D 228
      004D 229      ;+
      004D 230      ; Undefined exponentiation error - 0**0 or 0**(-n)
      004D 231      ; -
      004D 232
      004D 233      UNDEFINED:
      50 01 0F 79 004D 234      ASHQ      #15, #1, R0      ; R0/R1 = reserved floating operand
      7E 00'8F 9A 0051 235      MOVZBL      #MTH$K_UNDEXP, -(SP) ; Indicate undefined exponentiation.
      00000000'GF 01 FB 0055 236      CALLS      #1, G^MTH$$$SIGNAL ; convert to 32-bit condition code
      005C 237      ; and SIGNAL MTH$_UNDEXP
      005C 238      ; Note: 2nd arg not needed since no JSB OTSS$
      005C 239      ; is possible.
      04 005C 240      RET      ; return
      005D 241
      005D 242      ;+
      005D 243      ; The following handler is established to process exceptions which imply
      005D 244      ; underflow of the final result (floating overflow in either of the two MULG's
      005D 245      ; when exp < 0). On the occurrence of such an exception, the handler signals
      005D 246      ; MTH$_FLOUNDMAT.
      005D 247      ; -
      005D 248
      005D 249      EXC_HNDLR_UNDER:
      2B 001C 005D 250      .WORD      ^M<R2, R3, R4>      ; Entry mask
      005F 251      BSBB      SETUP      ; Set up R0:R3 and identify condition.
      0061 252      ; Return only if FLT0VF or FLTDIV.
      1E 04 A2 06 E1 0061 253      BBC      #PSL$V_FU, SF$W_SAVE_PSW(R2), CON_U ; Branch if caller has not enabled FU.
      54 00'8F 9A 0066 254      MOVZBL      #MTH$K_FLOUNDMAT, R4 ; Report MTH$_FLOUNDMAT, not SS$_FLT0VF.
      0C 11 006A 255      BRB      DO_SIG
      006C 256
      006C 257
      006C 258      ;+
      006C 259      ; The following handler is established to process exceptions which imply
      006C 260      ; overflow of the final result (floating overflow in either of the two MULG's
      006C 261      ; when exp > 0, floating overflow in the DIVG, or floating divide by zero in the
      006C 262      ; DIVG). On the occurrence of such an exception, the handler signals
      006C 263      ; MTH$_FLOOVEMAT.
      006C 264      ; -
      006C 265
      006C 266      EXC_HNDLR_OVER:
      1C 001C 006C 267      .WORD      ^M<R2, R3, R4>      ; Entry mask
      006E 268      BSBB      SETUP      ; Set up R0:R3 and identify condition.
      0070 269      ; Return only if FLT0VF or FLTDIV.
      50 01 0F 78 0070 270      ASHL      #15, #1, R0      ; Make the default result -0.0.
      54 00'8F 9A 0074 271      MOVZBL      #MTH$K_FLOOVEMAT, R4 ; Report MTH$_FLOOVEMAT, not SS$_FLTxxx.
      0078 272
      0078 273      DO_SIG: PUSHL      SF$L_SAVE_PC(R2) ; Report caller's PC, not exception PC.
      54 DD 007B 274      PUSHL      R4 ; Report MTH$_xxx, not SS$_xxx.
      00000000'GF 02 FB 007D 275      CALLS      #2, G^MTH$$$SIGNAL ; Signal the condition.
      0C A3 50 7D 0084 276      CON_U: MOVQ      R0, CH$SL_MCH_SAVR0(R3) ; If continued, restore R0 and R1.
      50 00' 00 0088 277      MOVL      S^#SS$_CONTINUE, R0 ; Continue from the original exception.
      04 008B 278      DO_RET: RET ; Exit from handler.

```



```

008C 279
008C 280 ;+
008C 281 ; Common setup routine for handlers. Returns normally if exception was FLT0VF,
008C 282 ; FLT0VF_F, FLTDIV, or FLTDIV_F. If the exception was anything else, it
008C 283 ; executes a RET, causing an exit from the handler with R0 = 0, which is
008C 284 ; equivalent to $$$_RESIGNAL. In the case of a normal return (FLT0VF, FLT0VF_F,
008C 285 ; FLTDIV, or FLTDIV_F) it sets up R0:R3 as follows:
008C 286 ; R0/R1: 0
008C 287 ; R2: address of establisher's frame
008C 288 ; R3: address of mechanism array
008C 289 ; -
008C 290
008C 291 SETUP: CLRQ R0 ; Set default result to 0.0.
52 04 AC 7D 008E 292 MOVQ CHF$L_SIGARGLST(AP), R2 ; R2 = address of signal array
0000'8F 04 A2 B1 0092 293 ; R3 = address of mechanism array
0092 294 CMPW CHF$L_SIG_NAME(R2), $$$_FLT0VF
0098 295 ; Was it a floating overflow trap?
0000'8F 04 A2 B1 0098 296 BEQL DO_RSB ; Branch if yes.
009A 297 CMPW CHF$L_SIG_NAME(R2), $$$_FLT0VF_F
00A0 298 ; Or a floating overflow fault?
0000'8F 04 A2 B1 00A0 299 BEQL DO_RSB ; Branch if yes.
00A2 300 CMPW CHF$L_SIG_NAME(R2), $$$_FLTDIV
00A8 301 ; Or a floating divide by zero trap?
0000'8F 04 A2 B1 00A8 302 BEQL DO_RSB ; Branch if yes.
00AA 303 CMPW CHF$L_SIG_NAME(R2), $$$_FLTDIV_F
00B0 304 ; Or a floating divide by zero fault?
00B0 305 BNEQ DO_RET ; None of the above: return from handler
00B2 306 ; with R0 = 0.
08 A2 97 AF 9E 00B2 307 DO_RSB: MOVAB B^POWGJX, CHF$L_SIG_NAME+4(R2)
52 04 A3 D0 00B7 308 ; Change return PC to POWGJX.
00B7 309 MOVL CHF$L_MCH_FRAME(R3), R2 ; R2 = address of establisher's frame
00BB 310 RSB ; Return.
00BC 311
00BC 312 .END

```


OTSSPOWGJ
Symbol table

- G REAL*8 ** INTEGER*4 power routine C 1

16-SEP-1984 01:59:33
6-SEP-1984 11:28:16

VAX/VMS Macro V04-00
[MTHRTL.SRC]OTSSPOWGJ.MAR;1

Page 8
(4)

```
BASE = 00000004
CHFSL_MCH_FRAME = 00000004
CHFSL_MCH_SAVRO = 0000000C
CHFSL_SIGARGLIST = 00000004
CHFSL_SIG_NAME = 00000004
CON U 00000084 R 02
DO_RET 0000008B R 02
DO_RSB 000000B2 R 02
DO_SIG 00000078 R 02
EXC_HNDLR_OVER 0000006C R 02
EXC_HNDLR_UNDER 0000005D R 02
EXP = 0000000C
EXPGTR 00000023 R 02
MTHSSIGNAL ***** X 00
MTHSK_FLOOVEMAT ***** X 00
MTHSK_FLOUNDMAT ***** X 00
MTHSK_UNDEXP ***** X 00
OTSSPOWGJ 00000000 RG 02
PARTIAL 00000033 R 02
POWGJX 0000004C R 02
PSLSV_FU = 00000006
SETUP 0000008C R 02
SFSL_SAVE_PC = 00000010
SFSW_SAVE_PSW = 00000004
SQUAR 00000027 R 02
SQUAR1 0000002C R 02
SSS_CONTINUE ***** X 00
SSS_FLTDIV ***** X 00
SSS_FLTDIV_F ***** X 00
SSS_FLTOVF ***** X 00
SSS_FLTOVF_F ***** X 00
UNDEFINED 0000004D R 02
```

+-----+
! Psect synopsis !
+-----+

PSECT name	Allocation	PSECT No.	Attributes															
. ABS .	00000000 (0.)	00 (0.)	NOPIC	USR	CON	ABS	LCL	NOSHR	NOEXE	NORD	NOWRT	NOVEC	BYTE					
\$ABSS	00000000 (0.)	01 (1.)	NOPIC	USR	CON	ABS	LCL	NOSHR	EXE	RD	WRT	NOVEC	BYTE					
_OTSSCODE	000000BC (188.)	02 (2.)	PIC	USR	CON	REL	LCL	SHR	EXE	RD	NOWRT	NOVEC	LONG					

+-----+
! Performance indicators !
+-----+

Phase	Page faults	CPU Time	Elapsed Time
Initialization	35	00:00:00.09	00:00:00.97
Command processing	128	00:00:00.73	00:00:03.98
Pass 1	137	00:00:02.04	00:00:08.13
Symbol table sort	0	00:00:00.09	00:00:00.16
Pass 2	68	00:00:00.92	00:00:03.90
Symbol table output	4	00:00:00.03	00:00:00.04
Psect synopsis output	2	00:00:00.02	00:00:00.02
Cross-reference output	0	00:00:00.00	00:00:00.00

OTSSPOWGJ
VAX-11 Macro Run Statistics

- G REAL*8 ** INTEGER*4 power routine^{D 1}

16-SEP-1984 01:59:33
6-SEP-1984 11:28:16

VAX/VMS Macro V04-00
[MTHRTL.SRC]OTSPOWGJ.MAR;1

Page 9
(4)

Assembler run totals 376 00:00:03.94 00:00:17.21

The working set limit was 1050 pages.

9091 bytes (18 pages) of virtual memory were used to buffer the intermediate code.

There were 10 pages of symbol table space allocated to hold 106 non-local and 0 local symbols.

372 source lines were read in Pass 1, producing 13 object records in Pass 2.

11 pages of virtual memory were used to define 10 macros.

+-----+
! Macro library statistics !
+-----+

Macro library name

Macros defined

_S255SDUA28:[SYSLIB]STARLET.MLB;2

6

148 GETS were required to define 6 macros.

There were no errors, warnings or information messages.

MACRO/ENABLE=SUPPRESSION/DISABLE=(GLOBAL,TRACEBACK)/LIS=LIS\$:OTSPOWGJ/OBJ=OBJ\$:OTSPOWGJ MSRC\$:MTHJACKET/UPDATE=(ENH\$:MTHJACKET)+MSRC

0264 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

0265 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY