


```

MM      MM      AAAAAA  KK      KK  EEEEEEEEE  RRRRRRRR  000000  000000  TTTTTTTTTT
MM      MM      AAAAAA  KK      KK  EEEEEEEEE  RRRRRRRR  000000  000000  TTTTTTTTTT
MMMM    MMMM    AA      AA  KK      KK  EEEEEEEEE  RR      RR  00      00  00      00  TT
MMMM    MMMM    AA      AA  KK      KK  EEEEEEEEE  RR      RR  00      00  00      00  TT
MM      MM      AA      AA  KK      KK  EEEEEEEEE  RR      RR  00      00  00      00  TT
MM      MM      AA      AA  KK      KK  EEEEEEEEE  RR      RR  00      00  00      00  TT
MM      MM      AA      AA  KKKKKK  EEEEEEEEE  RR      RR  00      00  00      00  TT
MM      MM      AA      AA  KKKKKK  EEEEEEEEE  RR      RR  00      00  00      00  TT
MM      MM      AAAAAAAAAA KK      KK  EEEEEEEEE  RR  RR  00      00  00      00  TT
MM      MM      AAAAAAAAAA KK      KK  EEEEEEEEE  RR  RR  00      00  00      00  TT
MM      MM      AA      AA  KK      KK  EEEEEEEEE  RR      RR  00      00  00      00  TT
MM      MM      AA      AA  KK      KK  EEEEEEEEE  RR      RR  00      00  00      00  TT
MM      MM      AA      AA  KK      KK  EEEEEEEEE  RR      RR  00      00  00      00  TT
MM      MM      AA      AA  KK      KK  EEEEEEEEE  RR      RR  00      00  00      00  TT
MM      MM      AA      AA  KK      KK  EEEEEEEEE  RR      RR  000000  000000  TT
MM      MM      AA      AA  KK      KK  EEEEEEEEE  RR      RR  000000  000000  TT

```

....
....
....
....

```

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
LLLLLLLLLL IIIIII  SSSSSSSS
LLLLLLLLLL IIIIII  SSSSSSSS

```

```

$ |
$ | MAKEROOT - Make additional system directories for cluster common
$ |           system disks.
$ |
$ | P1 - Name of directory to make (with or without brackets).
$ | P2 - SCSNODE and SCSSYSTEMID in the format NODE:ID (star:2208)
$ |
$ SAY = 'WRITE SYSS$OUTPUT'
$ ASK = 'INQUIRE /NOPUNCT'
$ F_LOG = '/LOG'
$ DEFAULT_PFSIZE = 4096
$ DEFAULT_SWAPSIZE = 4096
$ SD = 'SYSS$SYSDEVICE:'
$ IF F$TRNLNM('SYSS$SPECIFIC') .EQS. F$TRNLNM('SYSS$COMMON') THEN GOTO NOTCDISK
$ IF F$GETSYI('VAXCLUSTER') .EQ. 0 THEN GOTO NOTCLUSTER
$ ROOT = P1
$GET_ROOT:
$ IF ROOT .NES. '' THEN GOTO CHECK_ROOT
$ ASK ROOT 'What is the name of the new system root? '
$ GOTO GET_ROOT
$GET_ROOT1:
$ ROOT = ''
$ GOTO GET_ROOT
$CHECK_ROOT:
$ ROOT = ROOT - '[' - ']' - '<' - '>'
$ ROOT = F$EDIT(ROOT,'COLLAPSE,UPCASE')
$ IF ROOT .EQS. F$TRNLNM('SYSS$TOPSYS') THEN GOTO RUNNINGROOT
$ ! Syntax check the root, and see if it already exists
$ IF F$EXT(0,3,ROOT) .NES. 'SYS' THEN GOTO GET_ROOT1
$ IF F$LEN(ROOT) .NE. 4 THEN GOTO GET_ROOT1
$ T1 = F$EXT(3,1,ROOT)
$ IF (T1 .GES. '1') .AND. (T1 .LES. '9') THEN GOTO CKR10
$ IF (T1 .LTS. 'A') .OR. (T1 .GTS. 'D') THEN GOTO GET_ROOT1
$CKR10:
$ IF F$PARSE('''SD'[000000]''ROOT'.DIR') .EQS. '' THEN GOTO GET_ROOT1
$ IF F$SEARCH('''SD'[000000]''ROOT'.DIR') .NES. '' THEN GOTO ROOTXIST
$GETNEWPARAMS:
$ SCSNODE = ''
$ SCSSYSTEMID = ''
$ IF P2 .EQS. '' THEN GOTO GET_NODE
$ SCSNODE = F$ELEMENT(0,'',P2)
$ SCSSYSTEMID = F$ELEMENT(1,'',P2)
$GET_NODE:
$ IF SCSNODE .NES. '' THEN GOTO CHECK_NODE
$ ASK SCSNODE 'What is the nodename of the new node (SCSNODE)? '
$ GOTO GET_NODE
$CHECK_NODE:
$ T1 = F$LENGTH(SCSNODE)
$ IF T1 .GT. 6 THEN SCSNODE = ''
$ IF T1 .GT. 6 THEN GOTO GET_NODE
$GET_ID:
$ IF SCSSYSTEMID .NES. '' THEN GOTO CHECK_ID
$ ASK SCSSYSTEMID 'What is the SCS System id of the new node? '
$ GOTO GET_ID
$CHECK_ID:
$ T1 = F$INTEGER(SCSSYSTEMID)
$ IF T1 .EQ. 0 THEN SCSSYSTEMID = ''
$ IF T1 .EQ. 0 THEN GOTO GET_ID
$GET_PFSIZE:
$ ASK PFSIZE 'Size of pagefile for the new root [''DEFAULT_PFSIZE' blocks]? '
$ IF PFSIZE .EQS. '' THEN PFSIZE = DEFAULT_PFSIZE
$ T1 = F$INTEGER(PFSIZE)
$ IF T1 .EQ. 0 THEN GOTO GET_PFSIZE

```

```

$GET SWAPSIZE:
$ ASK SWAPSIZE "Size of swap file for the new root ['DEFAULT_SWAPSIZE' blocks]? "
$ IF SWAPSIZE .EQS. "" THEN SWAPSIZE = DEFAULT_SWAPSIZE
$ T1 = F$INTEGER(SWAPSIZE)
$ IF T1 .EQ. 0 THEN GOTO GET_SWAPSIZE
$DOIT:
$ SAY "Creating directory tree 'ROOT'..."
$ CD = "CREATE /DIR 'F LOG' /OWN=[1,4] /PROT=(S=RWE,O=RWE,G=RE,W=RE) 'SD'['ROOT'"
$ SDROOT = SD + "[" + ROOT + "]"
$ IF F$SEARCH("SD'[0,0]'"ROOT'.DIR') .EQS. "" THEN 'CD'] !Make the root directory
$ IF F$SEARCH("SDROOT'SYSEXE.DIR') .EQS. "" THEN 'CD'.SYSEXE]
$ IF F$SEARCH("SDROOT'SYSLIB.DIR') .EQS. "" THEN 'CD'.SYSLIB]
$ SET FILE /ENTER='SDROOT'SYSCOMMON.DIR -
      SYSS$SPECIFIC:[000000]SYSCOMMON.DIR; 'F LOG'
$ IF F$SEARCH("SDROOT'SYSTEEST.DIR') .EQS. "" THEN 'CD'.SYSTEEST]
$ IF F$SEARCH("SDROOT'SYSMAINT.DIR') .EQS. "" THEN 'CD'.SYSMAINT]
$ IF F$SEARCH("SDROOT'SYSMGR.DIR') .EQS. "" THEN 'CD'.SYSMGR]
$ IF F$SEARCH("SDROOT'SYSHLP.DIR') .EQS. "" THEN 'CD'.SYSHLP]
$ IF F$SEARCH("SD'['ROOT'SYSHLP]EXAMPLES.DIR') .EQS. "" THEN 'CD'.SYSHLP.EXAMPLES]
$ IF F$SEARCH("SDROOT'SYSUPD.DIR') .EQS. "" THEN 'CD'.SYSUPD]
$ IF F$SEARCH("SDROOT'SYSMSG.DIR') .EQS. "" THEN 'CD'.SYSMSG]
$ IF F$SEARCH("SDROOT'SYSERR.DIR') .EQS. "" THEN 'CD'.SYSERR]
$ IF F$SEARCH("SDROOT'SYSCBI.DIR') .EQS. "" THEN 'CD'.SYSCBI]
$ SAY "XUPGRADE-I-CREATED, System root 'ROOT' created."
$ SAY "XUPGRADE-I-CREPAGE$WAP, Creating page and swap files for new root."
$ SDROOT = SDROOT - "]"
$ SYSGEN = "$SYSGEN"
$ IF F$SEARCH("SDROOT'.SYSEXE]PAGEFILE.SYS') .EQS. "" THEN GOTO CP10
$ IF F$FILE("SDROOT'.SYSEXE]PAGEFILE.SYS','ALQ') .GE. PFSIZE THEN GOTO CP20
$CP10:
$ SYSGEN CREATE 'SDROOT'.SYSEXE]PAGEFILE.SYS/SIZE='PFSIZE'
$CP20:
$ IF F$SEARCH("SDROOT'.SYSEXE]SWAPFILE.SYS') .EQS. "" THEN GOTO CP30
$ IF F$FILE("SDROOT'.SYSEXE]SWAPFILE.SYS','ALQ') .GE. SWAPSIZE THEN GOTO CP40
$CP30:
$ SYSGEN CREATE 'SDROOT'.SYSEXE]SWAPFILE.SYS/SIZE='SWAPSIZE'
$CP40:
$ OPEN /WRITE F1 'SDROOT'.SYSEXE]MODPARAMS.DAT
$ WRITE F1 ""
$ WRITE F1 "Site specific AUTOGEN data file. In a VAXcluster where a common system"
$ WRITE F1 "disk is being used, this file should reside in SYSS$SPECIFIC:[SYSEXE],"
$ WRITE F1 "not a common system directory."
$ WRITE F1 ""
$ WRITE F1 "Add modifications that you wish to make to AUTOGEN's hardware configuration"
$ WRITE F1 "data, system parameter calculations, and page, swap, and dump file sizes to "
$ WRITE F1 "the bottom of this file."
$ WRITE F1 ""
$ WRITE F1 "PAGEFILE='PFSIZE'"
$ WRITE F1 "SWAPFILE='SWAPSIZE'"
$ CLOSE F1
$ COPY SYSS$COMMON:[SYSMGR]VMSIMAGES.DAT 'SD'['ROOT'.SYSMGR]
$ OPEN /WRITE F1 SYSS$SCRATCH:MAKEROOT.TMP
$ WRITE F1 "$ RUN SYSS$SYSTEM:SYSGEN"
$ WRITE F1 "USE CURRENT"
$ WRITE F1 "SET SC$NODE ""SC$NODE""
$ WRITE F1 "SET SC$SYSTEMID ""SC$SYSTEMID""
$ WRITE F1 "WRITE ""SDROOT'.SYSEXE]VAXVMS$SYS.PAR"
$ WRITE F1 "EXIT"
$ WRITE F1 "$ EXIT 1"
$ CLOSE F1
$ @SYSS$SCRATCH:MAKEROOT.TMP
$ DELETE SYSS$SCRATCH:MAKEROOT.TMP;*
$ TYPE SYSS$INPUT

```

Now you must build console media for the new system. This can be done in the following manner:

First create a scratch directory and SET DEFAULT to it.

Invoke SYSSUPDATE:DXCOPY.COM. When it asks "Copy from console medium (Y/N)?", answer YES. When prompted for the files to copy, answer *.*. DXCOPY will copy all the files from the current system console to the scratch directory.

Edit DEFBOO.COM, and adjust the value deposited in R5 to contain the root name in the high 4 bits. For example, if you just created root SYSS, then you must deposit the value 50000000 into R5.

Then, DISMOUNT CSA1, and insert a scratch console media. Re-invoke SYSSUPDATE:DXCOPY.COM, and answer NO to "Copy from console medium (Y/N)?". When prompted for the list of files to copy, answer *.*. DXCOPY will copy all the files from the scratch directory to the console. After DXCOPY completes, DISMOUNT CSA1, and take the media to your new system.

You must now boot the target node into the newly-created root. Use a conversational bootstrap, as you must change the startup command procedure. Use the SYSBOOT commands:

```
SET /STARTUP SYSSSYSTEM:STARTUP.COM
SET STARTUP_P1 'MIN'
CONTINUE
```

Then, after the system has booted, login and invoke AUTOGEN:

```
@SYSSUPDATE:AUTOGEN GETDATA REBOOT
```

```
$ EXIT 1
$ROOTXIST:
$ SAY "%UPGRADE-E-ROOTEXISTS, Root 'ROOT' already exists."
$ ASK YN "Do you want to continue [Y]?"
$ IF YN .EQS. "" THEN YN = "Y"
$ IF YN THEN GOTO CLEANUP_ROOT
$ EXIT
$CLEANUP_ROOT:
$ SDROOT = SD + "[" + ROOT
$ IF F$SEARCH("'"SDROOT'.SYSEXE]MODPARAMS.DAT') .NES. "" THEN -
  DELETE 'SDROOT'.SYSEXE]MODPARAMS.DAT;*
$ IF F$SEARCH("'"SDROOT'.SYSEXE]VAXVMSSYS.PAR') .NES. "" THEN -
  DELETE 'SDROOT'.SYSEXE]VAXVMSSYS.PAR;*
$ IF F$SEARCH("'"SDROOT'.SYSMGR]VMSIMAGES.DAT') .NES. "" THEN -
  DELETE 'SDROOT'.SYSMGR]VMSIMAGES.DAT;*
$ IF F$SEARCH("'"SD['ROOT']SYSCOMMON.DIR') .NES. "" THEN -
  SET FILE /REMOVE 'SDROOT']SYSCOMMON.DIR;*
$ DEFAULT PFSIZE = 0
$ IF F$SEARCH("'"SDROOT'.SYSEXE]PAGEFILE.SYS') .NES. "" THEN -
  DEFAULT PFSIZE = F$FILE("'"SDROOT'.SYSEXE]PAGEFILE.SYS',"ALQ")
$ DEFAULT SWAPSIZE = 0
$ IF F$SEARCH("'"SDROOT'.SYSEXE]SWAPFILE.SYS') .NES. "" THEN -
  DEFAULT SWAPSIZE = F$FILE("'"SDROOT'.SYSEXE]SWAPFILE.SYS',"ALQ")
$ GOTO GETNEWPARAMS
$RUNNINGROOT:
$ SAY "%UPGRADE-E-RUNNING, 'ROOT' is the root of the currently running system"
$ EXIT
$NOTCDISK:
```

\$ SAY "%UPGRADE-E-NOTDISK, This system disk is not set up as a ^{B 6} cluster system disk."
\$ EXIT
\$NOTCLUSTER:
\$ SAY "%UPGRADE-E-NOTCLUSTER, This node is not booted with VAXCLUSTER NE 0"
\$ EXIT

AUTOGEN LIS

NETCONFIG LIS

STARTUP LIS

SWAPFILES LIS

TRANSERR LIS

MAKEROOT LIS

SHUTDOWN LIS

SPKITBLD LIS

VMSINSTAL MEM

UINSTAL LIS

UUPDATE LIS

This page contains a grid of 100 small, illegible text fragments, likely representing a list of files or system components. The fragments are arranged in approximately 10 rows and 10 columns. Some fragments are larger and more legible than others, appearing as bolded headers or titles for sections of the list. The overall appearance is that of a dense, high-resolution scan of a document page.