

```

UUU          UUU  VVV          VVV  MMM          MMM  SSSSSSSSSSSSS
UUU          UUU  VVV          VVV  MMM          MMM  SSSSSSSSSSSSS
UUU          UUU  VVV          VVV  MMM          MMM  SSSSSSSSSSSSS
UUU          UUU  VVV          VVV  MMMMMM      MMMMMM  SSS
UUU          UUU  VVV          VVV  MMMMMM      MMMMMM  SSS
UUU          UUU  VVV          VVV  MMMMMM      MMMMMM  SSS
UUU          UUU  VVV          VVV  MMM          MMM  SSS
UUU          UUU  VVV          VVV  MMM          MMM  SSS
UUU          UUU  VVV          VVV  MMM          MMM  SSS
UUU          UUU  VVV          VVV  MMM          MMM  SSS
UUU          UUU  VVV          VVV  MMM          MMM  SSS
UUU          UUU  VVV          VVV  MMM          MMM  SSS
UUU          UUU  VVV          VVV  MMM          MMM  SSS
UUU          UUU  VVV          VVV  MMM          MMM  SSS
UUU          UUU  VVV          VVV  MMM          MMM  SSS
UUU          UUU  VVV          VVV  MMM          MMM  SSS
UUU          UUU  VVV          VVV  MMM          MMM  SSS
UUU          UUU  VVV          VVV  MMM          MMM  SSS
UUU          UUU  VVV          VVV  MMM          MMM  SSS
UUUUUUUUUUUUUUUUUU  VVV  VVV          MMM          MMM  SSSSSSSSSSSSS
UUUUUUUUUUUUUUUUUU  VVV  VVV          MMM          MMM  SSSSSSSSSSSSS
UUUUUUUUUUUUUUUUUU  VVV  VVV          MMM          MMM  SSSSSSSSSSSSS

```

RE  
S  
S  
E  
G  
S  
S  
V  
E

```

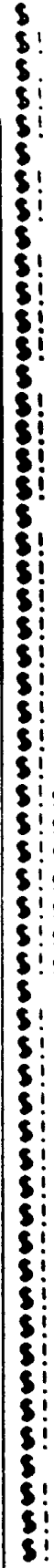
RRRRRRRR      EEEEEEEEEEE  MM      MM      000000      VV      VV      EEEEEEEEEEE
RRRRRRRR      EEEEEEEEEEE  MM      MM      000000      VV      VV      EEEEEEEEEEE
RR      RR      EE      MM      MMMM      00      00      VV      VV      EE
RR      RR      EE      MM      MMMM      00      00      VV      VV      EE
RR      RR      EE      MM      MM      00      00      VV      VV      EE
RR      RR      EE      MM      MM      00      00      VV      VV      EE
RRRRRRRR      EEEEEEEEEEE  MM      MM      00      00      VV      VV      EEEEEEEEEEE
RRRRRRRR      EEEEEEEEEEE  MM      MM      00      00      VV      VV      EEEEEEEEEEE
RR      RR      EE      MM      MM      00      00      VV      VV      EE
RR      RR      EE      MM      MM      00      00      VV      VV      EE
RR      RR      EE      MM      MM      00      00      VV      VV      EE
RR      RR      EE      MM      MM      00      00      VV      VV      EE
RR      RR      EEEEEEEEEEE  MM      MM      000000      VV      VV      EEEEEEEEEEE
RR      RR      EEEEEEEEEEE  MM      MM      000000      VV      VV      EEEEEEEEEEE

```

```

CCCCCCCC      000000      MM      MM
CCCCCCCC      000000      MM      MM
CC      00      00      MMMM      MMMM
CC      00      00      MMMM      MMMM
CC      00      00      MM      MM
CC      00      00      MM      MM
CC      00      00      MM      MM
CC      00      00      MM      MM
CC      00      00      MM      MM
CC      00      00      MM      MM
CC      00      00      MM      MM
CC      00      00      MM      MM
CCCCCCCC      000000      MM      MM
CCCCCCCC      000000      MM      MM

```



```

$
$ REMOVE.COM
$
$ This command procedure uses as input a data file (specified in P1 or
$ defaulted to SYSSUPDATE:VMSKITBLD.DAT on the system disk). The data file is
$ read as input commands to be executed by the procedure. The format of each
$ record is defined in VMSKITBLD.COM. P2 specifies the category of files to
$ be removed from the system disk. This is one of the options specified by
$ the UVMS qualifier in the data file. If P2 is not specified, then the user
$ is prompted for the option category.
$
$START:
$ ON ERROR THEN GOTO ALL_DONE
$ ON CONTROL Y THEN GOTO ALL_DONE
$ DEFDIR:='F$DIRECTORY()'
$ DEFUIC:='F$USER()'
$ DEFROOT := "SYSSSYSROOT:"
$ DEFEXE := "SYSSSYSROOT:[SYSEXE]"
$ DEFUPD := "SYSSSYSROOT:[SYSUPD]"
$ WRITE SYSSOUTPUT ""
$ WRITE SYSSOUTPUT -
$ "This command procedure removes entire options or sub-options
$ WRITE SYSSOUTPUT -
$ "of the MicroVMS kit. Help may be obtained for any of the prompts"
$ WRITE SYSSOUTPUT "by typing HELP."
$ WRITE SYSSOUTPUT ""
$ WRITE SYSSOUTPUT "Valid options and sub-options are:"
$ WRITE SYSSOUTPUT ""
$
$ Get data file name
$
$ FILENAME = P1
$ IF FILENAME .EQS. "" THEN FILENAME = "SYSSUPDATE:VMSKITBLD.DAT"
$ FILE = FILENAME
$
$ Find the valid values for OPTIONS and SUB-OPTIONS from the data file.
$
$ KINDA_VALID = "ALL,BASE,HELP,IGNORE"
$ VALID_OPTION = ""
$ PREV_VALID = ""
$ PREV_SUB = ""
$ OPEN/READ DATAF FILENAME
$ ON ERROR THEN GOTO CLOSE_DATAF
$ ON CONTROL Y THEN GOTO CLOSE_DATAF
$VALID_REQUESTS:
$ READ/END OF FILE=NORMAL EXIT DATAF RECORD
$ IF F$EXT(0,1,RECORD) .EQS. "$" THEN GOTO END_VALID
$ IF F$EXT(1,1,RECORD) .NES. "*" THEN GOTO VALID_REQUESTS
$ VALID = F$EXT(F$LOC("!* ",RECORD)+3,F$LOC(" ",RECORD)-3,RECORD)
$ IF VALID .NES. PREV_VALID THEN VALID_OPTION = VALID_OPTION + VALID + ","
$ IF VALID .NES. PREV_VALID THEN PREV_SUB = ""
$ IF VALID .NES. PREV_VALID THEN WRITE SYSSOUTPUT " ",VALID
$ PREV_VALID = VALID
$ VALID_SUB = F$EXT(F$LOC(" ",RECORD) + 1,1,RECORD)
$ VALID_VALID = PREV_SUB + VALID_SUB
$ PREV_SUB = VALID_VALID

```

```

$ WRITE SYSS$OUTPUT " " RECORD
$ GOTO VALID_REQUESTS
$END_VALID:
$ CLOSE DATAF
$ ON ERROR THEN GOTO ALL_DONE
$ ON CONTROL_Y THEN GOTO ALL_DONE
$ VALID_DEL = " "
$ VALID_OPTION = VALID_OPTION + KINDA_VALID
$
$ OPTIONS = P2
$ SAVE_OPTIONS = OPTIONS
$ SUBOPTS = ""
$ SUB_OPTION = ""
$ SUB_OPT_RETURN := ALL_DONE
$ REMOVED_OPTIONS = ""
$ ENT_OPT = P3
$
$ : If option(s) not specified then get options list
$
$GET_OPTION:
$ IF OPTIONS .NES. "" THEN GOTO VALIDATE_OPTION
$ WRITE SYSS$OUTPUT " "
$ IF REMOVED_OPTIONS .NES. "" -
$ THEN WRITE SYSS$OUTPUT 'Option(s) previously removed: ',REMOVED_OPTIONS,' '
$ WRITE SYSS$OUTPUT " "
$ WRITE SYSS$OUTPUT 'What layered VMS option(s) do you want to remove.'
$ INQUIRE OPTIONS 'Options must be separated by commas, e.g., USER,UTIL,SYSP'
$ SAVE_OPTIONS = OPTIONS
$ GOTO GET_OPTION
$
$ : Extract and validate an option, update options list
$
$VALIDATE_OPTION:
$ RETURN := GET_OPTION
$ OPTION = OPTIONS
$ IF F$LOC(", ",OPTIONS) .NE. F$LEN(OPTIONS) THEN OPTION = F$EXT(0,F$LOC(", ",OPTIONS),OPTIONS)
$EX_OPTION:
$ TEMP_1 = F$LOCATE(OPTION,VALID_OPTION)
$ IF (F$EXT(TEMP_1-1,1,VALID_OPTION) .NES. " ") THEN GOTO BAD_OPTION
$ IF TEMP_1 .EQ. F$LENGTH(VALID_OPTION) THEN GOTO BAD_OPTION
$ TEMP_2 = F$EXTRACT(TEMP_1,132,VALID_OPTION)
$ TEMP_2 = F$EXTRACT(0,F$LOCATE(", ",TEMP_2),TEMP_2)
$ IF OPTION .NES. TEMP_2 THEN GOTO BAD_OPTION
$ IF OPTION .EQS. 'HELP' THEN GOTO HELP_OPTION
$ IF OPTION .EQS. 'ALL' THEN GOTO BAD_OPTION
$ IF OPTION .EQS. 'BASE' THEN GOTO BAD_OPTION
$ OPTIONS = OPTIONS - OPTION - " "
$ OPT_RETURN := ALL_DONE
$ IF OPTIONS .NES. "" THEN OPT_RETURN := GET_OPTION
$ TOTAL_OPT = OPTION
$ REMOVED_OPTIONS = REMOVED_OPTIONS + " " + OPTION
$ REMOVED_SUB_OPTIONS = ""
$
$ : Remove check - entire option or not
$

```

```

$ENT_OPTION:
$   WRITE SYSS$OUTPUT " "
$   IF ENT_OPT .EQS. 'HELP' THEN GOTO HELP_ENT_OPT
$   IF ENT_OPT THEN GOTO CHECK_SEARCH
$   IF .NOT. ENT_OPT THEN GOTO GET_SUB_OPTION
$   INQUIRE ENT_OPT 'Do you want to remove the entire 'OPTION' option (y/n)'
$   IF ENT_OPT .EQS. 'Y' THEN GOTO ENT_OPTION
$   GOTO ENT_OPTION
$
$
$   : If not entire option then get sub-options list
$
$GET_SUB_OPTION:
$   RETURN := GET_SUB_OPTION
$   IF SUBOPTS .NES. ' ' THEN GOTO EX_SUB_OPT
$   WRITE SYSS$OUTPUT " "
$   IF REMOVED_SUB_OPTIONS .NES. ' ' -
$       THEN WRITE SYSS$OUTPUT 'Sub-option(s) previously removed: ',REMOVED_SUB_OPTIONS,' '
$   WRITE SYSS$OUTPUT " "
$   WRITE SYSS$OUTPUT 'What sub-option(s) do you want removed for option ',OPTION,'.'
$   INQUIRE SUBOPTS 'Sub-options must be separated by commas, e.g., A,C,F'
$   SAVE SUBOPTS = SUBOPTS
$   IF SUBOPTS .EQS. ' ' THEN GOTO GET_SUB_OPTION
$   IF SUBOPTS .EQS. 'HELP' THEN GOTO HELP_SUB_OPTION
$
$   : Extract and validate sub-option, update sub-options list
$
$EX_SUB_OPT:
$   IF SUBOPTS .EQS. ' ' THEN GOTO ALL_DONE
$   SUB_OPTION = SUBOPTS
$   IF F$LOC(' ',SUBOPTS) .NE. F$LEN(SUBOPTS) THEN SUB_OPTION = F$EXT(0,F$LOC(' ',SUBOPTS),SUBOPTS)
$VALIDATE_SUB_OPTIONS:
$   SUBOPTS = SUBOPTS - SUB_OPTION - ' '
$   SUB_OPT_RETURN = OPT_RETURN
$   IF SUBOPTS .NES. ' ' THEN SUB_OPT_RETURN := GET_SUB_OPTION
$   VALID_OPT = VALID 'OPTION'
$   IF F$LOCATE(SUB_OPTION,VALID_OPT) .EQ. -
$       F$LENGTH(VALID_OPT) THEN GOTO BAD_SUB_OPTION
$
$   TOTAL_OPT = OPTION + ' ' + SUB_OPTION
$   REMOVED_SUB_OPTIONS = REMOVED_SUB_OPTIONS + ' ' + SUB_OPTION
$
$   : Is the SEARCH utility installed?
$
$CHECK_SEARCH:
$   SRCH = F$SEARCH(' 'DEFEXE'SEARCH.EXE')
$
$   : If search for search command successful, then use search command
$   : on VMSKITBLD.DAT to create smaller file containing list of files
$   : to be deleted.
$
$OPT_SRCH:
$   IF S .EQS. 'Y' THEN GOTO OPEN_RECORD
$   SEAR 'FILENAME' 'UVMS=ALL', 'UVMS=' TOTAL_OPT-

```

```

      /OUTPUT='DEFUPD'UVOPTION.DAT
      FILE = '''DEFUPD'UVOPTION.DAT''
$
$
$ : This is the main loop that is executed for each record in the data file.
$ : If search command used then FILE opened is UVOPTION.DAT.  If search
$ : command not used then FILE opened is VMSKITBLD.DAT.
$
$OPEN_RECORD:
$ OPEN/READ DATAF 'FILE
$ ON ERROR THEN GOTO CLOSE_DATAF
$ ON CONTROL_Y THEN GOTO (CLOSE_DATAF
$RECORD_LOOP:
$ READ/END OF FILE=NORMAL EXIT DATAF RECORD
$ IF F$EXTRACT(0,1,RECORD) .EQS. '$' THEN GOTO RECORD_LOOP
$ IF F$EXTRACT(0,1,RECORD) .EQS. '!' THEN GOTO RECORD_LOOP
$ IF F$EXTRACT(0,1,RECORD) .EQS. ',' THEN GOTO RECORD_LOOP
$GET_UVMS:
$ UVMS = F$EXTRACT(F$LOCATE('UVMS=',RECORD)+5,132,RECORD)
$ IF UVMS .NES. '' THEN GOTO PARSE_UVMS
$ WRITE SYSS$OUTPUT '%WARNING - No UVMS qualifier on record: '
$ WRITE SYSS$OUTPUT '*****RECORD*****'
$ GOTO RECORD_LOOP
$
$PARSE_UVMS:
$ UVMS = F$EXTRACT(0,(F$LOCATE(' ',UVMS)),UVMS)
$ IF ENT_OPT THEN TOTAL_OPT = UVMS
$ SAVESET_NAME = UVMS
$ UVMS = F$EXTRACT(0,(F$LOCATE(' ',UVMS)),UVMS)
$ IF F$LOCATE(UVMS,VALID_OPTION) .NE. F$LENGTH(VALID_OPTION) THEN GOTO CHECK_OPTION
$ WRITE SYSS$OUTPUT '%WARNING - Invalid UVMS qualifier on record:'
$ WRITE SYSS$OUTPUT '*****RECORD*****'
$ GOTO RECORD_LOOP
$
$ : Check if this file is in the option we are deleting.
$
$CHECK_OPTION:
$ IF SAVESET_NAME .EQS. 'ALL' THEN GOTO GET_NAME
$ IF SAVESET_NAME .NES. TOTAL_OPT THEN GOTO RECORD_LOOP
$
$ : Get the name from the current record. (The name is always the first field
$ : in the record, and doesn't have a keyword associated with it.)
$
$GET_NAME:
$ NAME = F$EXTRACT(0,F$LOCATE(' ',RECORD),RECORD)
$
$ : Get the operation.
$
$GET_OPERATION:
$ OPERATION = F$EXTRACT(F$LOCATE('OP=',RECORD)+3,132,RECORD)
$ IF OPERATION .EQS. '' THEN GOTO NORMAL
$ OPERATION = F$EXTRACT(0,F$LOCATE(' ',OPERATION),OPERATION)
$ GOTO 'OPERATION
$
$SET:
$ SOURCE_DEFAULT = '[' + '''NAME'' + ']'
$CREATE:

```

```
SENDER:
$RENAME_C:
$RENAME_B:
$SYSGEN:
$   GOTO RECORD_LOOP
$
$BUILD:
$COPY:
$NCRMAL:
$CONTIG:
$   IF F$SEARCH(''DEFROOT''SOURCE_DEFAULT''NAME'') .EQS. '' THEN GOTO RECORD_LOOP
$   DELETE/LOG 'DEFROOT''SOURCE_DEFAULT''NAME';*
$   IF NAME .EQS. 'STARLET.OLB' THEN LIBR/CREATE 'DEFROOT''SOURCE_DEFAULT''NAME'
$   GOTO RECORD_LOOP
$
$NO_DATAF:
$   WRITE SYSS$OUTPUT '' Unable to open data parameter file.''
$   GOTO ALL_DONE
$
$HELP_ENT_OPT:
$   WRITE SYSS$OUTPUT ''
$   WRITE SYSS$OUTPUT ''Typing Y indicates you want the entire option''
$   WRITE SYSS$OUTPUT ''to be removed. Typing N indicates you want only''
$   WRITE SYSS$OUTPUT ''portions of the option to be removed. The prompt''
$   WRITE SYSS$OUTPUT ''for these sub_options follows if you type N.''
$   WRITE SYSS$OUTPUT ''
$   ENT_OPT = ''
$   GOTO ENT_OPTION
$BAD_OPTION:
$   WRITE SYSS$OUTPUT '' ''
$   WRITE SYSS$OUTPUT ''You specified an invalid option in: ',SAVE_OPTIONS,'''
$   WRITE SYSS$OUTPUT ''
$HELP_OPTION:
$   WRITE SYSS$OUTPUT ''The valid options are:
$   WRITE SYSS$OUTPUT ''
$   VALID = VALID_OPTION - KINDA_VALID
$BACK:
$   TEMP = F$EXT(0,F$LOC('',',VALID),VALID)
$   WRITE SYSS$OUTPUT ''           ''TEMP
$   VALID = VALID - TEMP - ''
$   IF VALID .NES. '' THEN GOTO BACK
$   OPTIONS = ''
$   WRITE SYSS$OUTPUT ''
$   WRITE SYSS$OUTPUT ''Options must be separated by commas.''
$   WRITE SYSS$OUTPUT ''
$   GOTO RETURN
$BAD_SUB_OPTION:
$   WRITE SYSS$OUTPUT '' ''
$   WRITE SYSS$OUTPUT ''You specified an invalid sub-option in: ',SAVE_SUBOPTS,'''
$HELP_SUB_OPTION:
$   WRITE SYSS$OUTPUT ''
$   WRITE SYSS$OUTPUT ''The valid sub-options for ',OPTION,' are:''
$   WRITE SYSS$OUTPUT '' ''
$   IF SRCH .NES. '' THEN GOTO SRCH_SUB_OPT_HELP
$   OPEN/READ DATAF 'FILENAME
$   ON ERROR THEN GOTO CLOSE_DATAF
```

```

$ ON CONTROL_Y THEN GOTO CLOSE_DATAF
$HELP_READ:
$ READ/END OF FILE=HELP_FINISH DATAF RECORD
$ IF F$EXTRACT(0,1,RECORD) .NES. '' THEN GOTO HELP_FINISH
$ IF F$EXTRACT(1,1,RECORD) .NES. '*' THEN GOTO HELP_READ
$ VALID = F$EXT(F$LOC('!* ',RECORD)+3,F$LOC(' ',RECORD)-3,RECORD)
$ IF VALID .EQS. OPTION THEN WRITE SYSS$OUTPUT-' ',RECORD
$ GOTO HELP_READ
$HELP_FINISH:
$ CLOSE DATAF
$ ON ERROR THEN GOTO ALL_DONE
$ ON CONTROL_Y THEN GOTO ALL_DONE
$ GOTO HELP_CONT
$SRCH_SUB_OPT_HELP:
$ STRING = '***** + '* ' + OPTION + ' ' + '*****'
$ SEARCH 'FILENAME 'STRING
$HELP_CONT:
$ WRITE SYSS$OUTPUT ''
$ WRITE SYSS$OUTPUT "Sub-options must be separated by commas."
$ SUBOPTS := ''
$ GOTO RETURN
$NORMAL_EXIT:
$ CLOSE DATAF
$ ON ERROR THEN GOTO ALL_DONE
$ ON CONTROL_Y THEN GOTO ALL_DONE
$ IF TOTAL_OPT .NES. (OPTION+' ' + SUB_OPTION) -
$ THEN TOTAL_OPT = OPTION
$ WRITE SYSS$OUTPUT ''
$ WRITE SYSS$OUTPUT "***** ',TOTAL_OPT,' REMOVED *****"
$ WRITE SYSS$OUTPUT ''
$ GOTO SUB_OPT_RETURN
$CLOSE_DATAF:
$ CLOSE DATAF
$ALL_DONE:
$ IF F$SEARCH('DEFUPD'UVOPTION.DAT') .NES. '' THEN DELETE 'DEFUPD'UVOPTION.DAT;*
$ SET DEFAULT 'DEFDIR'
$ EXIT

```

MOD

S  
A  
P  
S  
D  
DUM  
VMS  
VMS  
KFI  
GBL  
GBL  
MAX  
SYS  
IF  
IF  
BAL  
IRP  
IRP  
WSM  
NPA  
NPA  
PAG  
SPT  
LRP  
LRP  
SRP  
SRP  
MPW  
MPW  
SWP  
FRE  
FRE  
GRO  
BOR  
CLI  
LOC  
LOC  
RES  
DEA  
SCS  
SCS  
SCS  
SCS  
PAM  
PAP  
CJF  
TTY  
RMS  
ACF  
ACF  
ACF  
ACF



