


```

SSSSSSSS Pppppppp KK      KK      Iiiiiii TTTTTTTTTT BBBBBBBBBB LL      DDDDDDDD
SSSSSSSS Pppppppp KK      KK      Iiiiiii TTTTTTTTTT BBBBBBBBBB LL      DDDDDDDD
SS      PP      PP      KK      KK      II      TT      BB      BB      LL      DD      DD
SS      PP      PP      KK      KK      II      TT      BB      BB      LL      DD      DD
SS      PP      PP      KK      KK      II      TT      BB      BB      LL      DD      DD
SSSSSS Pppppppp KKKKKK      II      TT      BBBBBBBBBB LL      DD      DD
SSSSSS Pppppppp KKKKKK      II      TT      BBBBBBBBBB LL      DD      DD
SS      PP      KK      KK      II      TT      BB      BB      LL      DD      DD
SS      PP      KK      KK      II      TT      BB      BB      LL      DD      DD
SS      PP      KK      KK      II      TT      BB      BB      LL      DD      DD
SSSSSSSS PP      KK      KK      Iiiiiii TTT      TT      BBBBBBBBBB LLLLLLLLLL DDDDDDDD
SSSSSSSS PP      KK      KK      Iiiiiii TTT      TT      BBBBBBBBBB LLLLLLLLLL DDDDDDDD

```

```

LL      Iiiiiii SSSSSSSS
LL      Iiiiiii 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
LLLLLLLLLL Iiiiiii SSSSSSSS
LLLLLLLLLL Iiiiiii SSSSSSSS

```

5

IDENT V03-002

++
Facility: SPKITBLD, Procedure to Build a VMS Software Product Kit

Abstract: This procedure is used to build an installation kit for a VAX/VMS software product. It is intended to build kits which are installed by the "new" VMSINSTAL.

Parameters: P1 - product to be built
(all optional) P2 - device on which to build kit
P3 - files to place in first saveset

Author: Paul C. Anagnostopoulos, Creation Date: 30 June 1982

Modifications:

V03-002 HWS0099 Harold Schultz 06-Aug-1984
Fix label truncation when dealing with console devicesV03-001 HWS0094 Harold Schultz 24-Jul-1984
Fix spelling of 'resetting'
Allow longer facility names (up to 39 char., including vvu)--
on control_y then goto bad_kit
on warning then goto bad_kit
ask = "inquire/nopunctuation"
say = "write sys\$output"
remount = 0
say ""f\$fa0('!//!/_VAX/VMS V4.0 Software Product Kit Build Procedure!//')"p = "CMKRNL,LOG IO,SYSNAM,VOLPRO"
saved_privs = f\$setprv(p)
if .not. f\$priv(p) then say "You need the following privileges: ",p
if .not. f\$priv(p) then goto bad_kitif p1 .nes. "" then goto m5
type sys\$input:Kits have names in the format facvvu, where fac is the facility name,
vv is the major version number, and u is the update number.

ask p1 "Which kit would you like to build: "

\$m5: facility = f\$ext(0,f\$len(p1)-3,p1)
version = 'f\$ext(f\$len(p1)-3,2,p1)
update = 'f\$ext(f\$len(p1)-1,1,p1)

dest = p2

if dest .eqs. "" then ask dest "On which device is the kit to be built: "

if f\$loc(":",dest) .eq. f\$len(dest) then dest = dest + ":"

device = f\$parse(dest,,,"DEVICE","SYNTAX ONLY")

destdir = f\$parse(dest,"[0,0]","DIRECTORY","SYNTAX ONLY")

dest = device + destdir

if f\$loc("CSA",device) .ne. f\$len(device) then goto m7

if f\$getdvi(dest,"EXISTS") then -

if f\$getdvi(dest,"DEVCLASS") .eq. 1 .or. -

f\$getdvi(dest,"DEVCLASS") .eq. 2 then goto m7

say "Device must be a disk or tape."

goto bad_kit

\$m7:

remount = f\$loc("CSA",device) .ne. f\$len(device)

if .not. remount then goto m9

s = "\$sysgen"

s connect console

```

$ if f$getdvi(device,'MNT') then dismount 'device
$ allocate 'device
$m9:
$
$ if f$getdvi(device,'DEVCLASS') .eq. 1 .and. -
$ f$getdvi(device,'MNT') .and. -
$ not. f$getdvi(device,'FOR') then goto m15
$ if f$getdvi(device,'MNT') .and. f$getdvi(device,'FOR') .and. -
$ f$getdvi(device,'DEVCLASS') .eq. 2 then goto m_tape
$ if f$getdvi(device,'MNT') then dismount/nounload 'device
$ say "Please ready the first distribution volume on "device."
$m10: ask w "Are you ready?"
$ if .not. w then goto m10
$ mount/foreign/noassist 'device
$ remount = 1
$
$m15: if f$getdvi(device,'DEVCLASS') .eq. 2 then goto m_TAPE
$ if f$getdvi(device,'MAXBLOCK') .gt. 8*512 then goto m_DISK
$m_CONSOLE:
$ backup_qual = '/init/block=9000/group=25'
$ if f$len(facility) .gt. 10 then facility = f$ext(0,10,facility)
$ labels = "'facility'01,'facility'02,'facility'03,'facility'04,'facility'05,"
$ labels = labels + "'facility'06,'facility'07,'facility'08,'facility'09,'facility'10"
$ goto m20
$m_DISK:
$ backup_qual = '/block=9000/group=25'
$ labels = facility
$ lim = 12
$ goto m19
$m_TAPE:
$ backup_qual = '/rewi/dens=1600'
$ ask w "Are there already savesets on this tape? "
$ if w then backup_qual = '/norewi/dens=1600'
$ labels = facility
$ lim = 6
$ goto m19
$m19:
$ if f$len(labels) .le. lim then goto m20
$ labels = f$ext(0,lim,labels)
$m20:
$ say "Saveset A must contain the KITINSTAL.COM procedure."
$ letters = "ABCDEFGHIJKLMNOPQRSTUVWXYZ"
$ ssn = 0
$
$s10:
$ ssn = ssn + 1
$ ssnname = f$ext(ssn-1,1,letters)
$ ssid = ssnname
$ if ssn .gt. 26 then ssnname = f$fao('VMI !4ZL',ssn)
$ if ssn .gt. 26 then ssid = f$fao('!4ZL',ssn)
$ say "'f$fao('!/_!_!AS V!UL!UL, Saveset !AS!/',facility,version,update,ssid)"
$ f = p3
$ if p3 .eqs. "" then ask f "Which files: "
$ prot = "(S:RWED,O:RWED,G:RWED,W:RE)"
$ write sys$output "Resetting protection on all files to ",prot,"..."
$ set protection=prot f
$s12: backup/comment="VAX/VMS SPKITBLD Procedure"/inter/log/verify -
$ f -
$ dest''p1.'ssnname/label=('labels)/save'backup_qual
$ if p2 .nes. "" then goto s19
$ ask w "Is there another saveset to create [NO]? "
$ if .not. w then goto s19

```

```

$
$
$ p3 = ""
$ backup_qual = backup_qual + "/noinit/norewi"
$ if f$getdvi(device,"DEVCLASS").eq. 2 .or. -
$   f$getdvi(device,"MAXBLOCK").gt. 8*512 then goto s10
$ ask w "Does it need to start on a new volume [NO]? "
$ if .not. w then goto s10
$ dismount/nounload 'device
$ mount/foreign/noassist/nomessage 'device
$ i = f$loc(f$getdvi(device,"VOLNAM"),labels)
$ labels = f$ext(i+f$len(facility)+3,999,labels)
$ if f$len(labels) .le. lim then goto s13
$ labels = f$ext(f$len(labels)-lim,lim,labels)
$ s13: backup_qual = backup_qual + "/init"
$ dismount 'device
$ say "Please mount the next distribution volume on ''device.'"
$ s15: ask w "Are you ready? "
$ if .not. w then goto s15
$ mount/foreign/noassist 'device
$ goto s10
$
$ s19:
$
$ say ""'f$fao('!/_Kit Build Completed Successfully!/_!/'")"
$
$ done:
$ on warning then exit
$ if .not. remount then goto d9
$ if f$getdvi(device,"MNT") then dismount/nounload 'device
$ if f$loc("CSA",device) .eq. f$len(device) then goto d9
$ say "Please remount the console volume."
$ d5: ask w "Are you ready? "
$ if .not. w then goto d5
$ mount/foreign/system/protection=s:rwlp/nowrite/noassist 'device console
$ d9: w = f$setpriv(saved_privs)
$ exit
$
$ bad_kit:
$ type sys$input:
$
$ +-----+
$ | THIS IS A BAD KIT |
$ +-----+
$
$ goto done

```

AUTOGEN
LIS

DISKCOMP
LIS

DISKPART
LIS

DISKSETUP
LIS

DISKSPACE
LIS

DISKUTIL
LIS

DISKVERIF
LIS

DISKWRITE
LIS

DISKINFO
LIS

DISKREPAIR
LIS

DISKTEST
LIS

DISKERR
LIS

DISKCHK
LIS

NETCONFIG
LIS

STARTUP
LIS

TIMESLIP
LIS

TIMEZONE
LIS

TIMESETUP
LIS

TIMEVARY
LIS

TIMECHG
LIS

TIMEDEL
LIS

TIMEADJ
LIS

TIMECALC
LIS

TIMECONV
LIS

TIMEOP
LIS

TIMEPRN
LIS

SWAPFILES
LIS

TRANSERR
LIS

MAKEROOT
LIS

SHUTDOWN
LIS

SPRITBLD
LIS

UMSINSTAL
LIS

UMSINSTAL
MEM

UMSUPDTE
LIS