CERN Program Library Long Writeup Z312

# VAXTAP

VAX/VMS

Tape Handling Package

Version 1.20 (December 1992)

Application Software Group

Computers and Network Division

CERN Geneva, Switzerland

CERN Program Library Office
CERN-CN Division
CH-1211 Geneva 23
Switzerland
Tel. +41 22 767 4951
Fax. +41 22 767 7155
Bitnet: CERNLIB@CERNVM
DECnet: VXCERN::CERNLIB (node 22.190)
Internet: CERNLIB@CERNVM.CERN.CH



*Contact Person*: Jamie Shiers /CN (JAMIE@CERNVM.CERN .CH)
*Technical Realization*: Michel Goossens /CN (GOOSSENS@CERNVM.C ERN.CH)

## Preliminary remarks

This **Complete Reference** of the VAXTAP package consists of three parts:

**1** An **overview** of the system.
**2** A **user guide**, describing each command in detail.
**3** An **installation and management guide**.

| Throughout this manual, commands to be **entered** are `underlined` |
|---|

## Acknowledgements

This document has been produced using LaTeX with the `CERNMAN` style option. All pictures shown are produced with PAW and are included in PostScript[?] format in the manual.

## Related Documents

This document can be complemented by the following documents:

- The FATMEN Report - DD/89/15[2]
- FATMEN User Guide [3]
- TMS - The CERN Tape Management System (to be published)

## Table of Contents

## List of Figures

## List of Tables

# Part I

# VAXTAP – Overview

# Chapter 1: Introduction

The utilities described in this manual were originally developed for the CERN VAXcluster VXCERN.

Several essential features were felt to be missing from standard VAX tape handling utlities. The most important of these were

- Support for IBM SL tape labels
- Support for visual identifiers (VIDs)
- Efficient device allocation

To this end, a new utility was written, known as SETUP. This command, which uses standard VAX/VMS system services, performs device allocation, tape mounting and label checking.

This command was soon joined by the other utilities described in this manual, namely LABELDUMP, STAGE, TAPECOPY, WRTAPE and XTAPE.

An interface to the CERN Tape Management System exists, and is activated at installation time as described in the installation chapter of this manual.

In addition, and interface from the FATMEN File and Tape Management system is also provided. Further details can be found in the FATMEN manual [3].

Originally, the commands described in this manual were defined using the DCL `SET COMMAND` utility. However, they have recently been rewritten so that this is no longer required, using the techniques described in [5].

# Part II

# VAXTAP – User Guide

# Chapter 2: VAXTAP help files for systems without a TMS interface

The following help files are for systems that do not have a TMS interface.

## 2.1 EINIT

```
1 EINIT

  Use the EINIT command to write a magnetic tape label encoded in EBCDIC.

  Format:

  RUN SETUP_EXE:EINIT

  The volume to be labelled must already be mounted /FOREIGN/WRITE on
  the tape unit with logical name INTAPE.

  The utility will prompt for the volume label to be written.
```

## 2.2 LABELDUMP

```
1 LABELDUMP


   Use the LABELDUMP command to find out the magnetically recorded label
   (if any) of a magnetic tape.

   Format:

   LABELDUMP [(visual identifier)]


2 Parameters
  (visual identifier)

  Specifies the number of the tape on the tape reel itself.
  If you do not specify this parameter, you are prompted for it.

2 Qualifiers

/GENERIC

   /GENERIC=TA90 (D)

 Use this qualifier to instruct LABELDUMP to allocate a certain
 device type. If not specified, the generic device will default to
 TA90 - the IBM 3480 compatible drive.
```

```
 The generic devices available on your system are defined by the system
 manager as logical names. For example, the logical name for TA90 drives
 is SETUP_TA90S. The following table gives some examples.

 Generic Device type     Characteristics     Logical name
 ===================     ===============     ============
 TA90 (or 3480)          3480 cartridge      SETUP_TA90S (or SETUP_3480S)
 TA78                    6250 bpi 9 track    SETUP_TA78S
 TK50                    TK50 cartridge      SETUP_TK50S

/DEBUG

 This qualifer requires SYSPRV and is primarily of use for debugging
 new versions of the LABELDUMP command. If specified, extra debug information
 will be displayed during command processing.

/WAIT

   /WAIT     (Default in BATCH)
   /NOWAIT   (Default otherwise)

  This parameter controls whether LABELDUMP will exit if all tape drives
  are currently in use, or wait until one becomes free. If executing
  in BATCH mode, /WAIT is in effect unless explicitly negated.


2 Examples

  LABELDUMP XIN101

  Dumps the label of XIN tape XIN101.
```

## 2.3 SETUP

```
1 SETUP

 The SETUP command should always be used to mount magnetic tapes,
 rather than the standard VMS MOUNT command.
 SETUP command will ALLOCATE a tape drive, MOUNT the tape
 and perform volume-label checking. The tape is then re-mounted via a
 standard VMS mount command and made available for user-processing.

 In addition to the qualifiers presented below, any MOUNT qualifers
 may be used. See HELP MOUNT for further information on additional
 qualifiers.


 Format:

      SETUP [(volume-label|VSN)] [(visual-identifier|VID)] [(logical-name)]

2 Parameters
```

```
 (volume-label|VSN)

  Specifies the MAGNETICALLY recorded label on the volume.
  Labels can have from 1 through 6 characters. This is the
  equivalent of the IBM MVS volume serial name (VSN) on the DD statement.

  The volume-label parameter is  always required. If your tape is unlabelled,
  specify the same value as the visual-identifier.

  N.B. if the VSN is in lower or mixed case, it must be enclosed in
  double quotes.

 (visual-identifier|VID)

  Specifies the contents of the sticky label on the volume reel itself.
  VID's must be exactly 6 characters.
  The visual-identifier must be a CERN tape number (5 digits followed
  be a check letter), an experimental tape number, (1 or 2 characters
  followed by 5 or 4 digits) or an XIN/XUT number.
  XIN tapes are read only.

  The visual-identifier parameter is  always required.

 logical-name[:]

  Defines a 1- through  63-alphanumeric  character  string  logical
  name to be associated with the device allocated.
  The logical name must be used to reference the drive on which the
  tape is mounted.

  If you do not specify a logical name, the SETUP  command  assigns
  the  default  logical  name  TAPE$volume-label.

2 Qualifiers
/BACKUP

 If you specify /BACKUP, your tape will be mounted /FOREIGN, after
 SETUP has verified that the correct volume-label has been specified.
 Use this qualifier to read and write tapes in BACKUP format, in
 addition to the relevant /LABEL qualifier.

/DEBUG

 This qualifer requires SYSPRV and is primarily of use for debugging
 new versions of the SETUP command. If specified, extra debug information
 will be displayed during command processing.

/FOREIGN

 Use of this qualifier requires VOLPRO or BYPASS privilege.
 Without privilege, tapes may only be mounted /FOREIGN if the
 /BACKUP qualifier is specified, together with the correct
 /LABEL qualifier and value. See the description of the /BACKUP
 and /LABEL qualifiers and the examples section for more details.

 If you do not know the magnetic label of a tape, use the LABELDUMP
 command to determine it.
```

```
/GENERIC

   /GENERIC=TA90 (D)

 Use this qualifier to instruct SETUP to allocate a certain
 device type. If not specified, the generic device will default to
 TA90 - the IBM 3480 compatible drive.

 The generic devices available on your system are defined by the system
 manager as logical names. For example, the logical name for TA90 drives
 is SETUP_TA90S. The following table gives some examples.

 Generic Device type     Characteristics     Logical name
 ===================     ===============     ============
 TA90 (or 3480)          3480 cartridge      SETUP_TA90S (or SETUP_3480S)
 TA78                    6250 bpi 9 track    SETUP_TA78S
 TK50                    TK50 cartridge      SETUP_TK50S

 N.B. if /GENERIC is NOT specified and SETUP has been installed with
 the TMS option, SETUP will obtain the correct device information from
 the TMS.

/LABEL

   /LABEL=ASCII  (D)
   /LABEL=EBCDIC
   /LABEL=NONE
   /NOLABEL

 Indicates, for tape volumes, whether the  tape  contains  VAX/VMS
 ANSI labels.  Note that /NOLABEL is equivalent to /LABEL=NONE.

 You should specify /LABEL=EBCDIC if you wish to read IBM SL tapes.
 However, data set positioning and translation must be performed by
 the user as the tape is positioned at beginning of tape by SETUP.
 To skip files, you may use the DCL command SET MAGTAPE/SKIP=FILES:n.
 Note that for labelled tapes the physical file number equals the
 logical file number times three minus one. See the EXAMPLES section
 for an example.

/WAIT

   /WAIT     (Default in BATCH)
   /NOWAIT   (Default otherwise)

  This parameter controls whether SETUP will exit if all tape drives
  are currently in use, or wait until one becomes free. If executing
  in BATCH mode, /WAIT is in effect unless explicitly negated.

/WRITE

   /WRITE
   /NOWRITE  (D)

 Controls   whether  the  volume  can  be  written.   /NOWRITE  is
 equivalent to write-locking the device.
```

```
2 Mounting_FOREIGN_tapes

On the CERN Computer Centre VAXcluster,tapes do not normally need
to be mounted /FOREIGN. The exception is for processing with BACKUP.

To mount a tape for processing by BACKUP (or any other program that
requires the tape to be mounted /FOREIGN), use

SETUP/BACKUP.

Note that you MUST specify the correct volume-label, if any.

If you want to MOUNT an UNLABELLED tape, use SETUP/NOLABEL.

If you want to MOUNT an IBM tape (i.e. LABELLED in EBCDIC),
use SETUP/LABEL=EBCDIC.


2 Examples
Setup a tape with labels encoded in ASCII.
$ .....
$ SETUP MX1015 MX1015             ! If the SETUP fails, the job will exit.
$ ASSIGN TAPE$MX1015 FOR001
$ RUN DSTANAL
$ EXIT

Setup a tape with labels encoded in EBCDIC.

$ ON ERROR THEN GOTO ABEND        ! Protect against problems with SETUP etc.
$ SETUP/NOWRITE/LABEL=EBCDIC 939927 XIN112 MYDRIVE
$ ASSIGN MYDRIVE:  FOR001         ! Processing of data sets is up to user.
$ RUN PROG
$ EXIT
$ !
$ABEND:
$ WRITE SYS$OUTPUT "Program abended"
$ EXIT

Mount an unlabelled tape

$ SETUP/NOLABEL XIN101 XIN101     ! Only if the tape is really unlabelled
$                                 ! will access to the tape be granted.
$ ASSIGN TAPE$XIN101 FOR010
$ RUN MYPROG
$ EXIT

Mount a labelled tape for writing to with BACKUP.

$SETUP/WRITE/BACKUP MICRO PR9902

Mount a tape written on a data acquisition VAX
with EBCDIC labels, blocksize = 3600 bytes and recordsize = 3600 bytes.

$SETUP/LABEL=EBCDIC/BLOCK=3600/RECORD=3600 MC1009 MC1009
$! Skip past volume label to logical file 1
$SET MAGTAPE/SKIP=FILES:1 TAPE$MC1009
$RUN MYPROG ! Uses EPIO, reading from unit 1.
```

## 2.4 STAGE

```
1 STAGE

Use the STAGE command to perform input or output staging of tape
data, clear existing staged data or query the status of staged
data.

Each group must be explicitly enabled to use tape staging. If you
find that your group is not enabled, please contact the UCO.

2 Command_Syntax

The syntax of the STAGE command is essentially that of SETUP, with some
extensions. Only these extensions are described below.

The format of the STAGE command is:

STAGE [(volume-label|VSN)] [(visual-identifier|VID)] [(logical-name)] -
  /NAME=fn/NUMBER=n

3 Parameters
 (volume-label|VSN)

Specifies the MAGNETICALLY recorded label on the volume.
Labels can have from 1 through 6 characters. This is the
equivalent of the IBM MVS volume serial name (VSN) on the DD statement.

The volume-label parameter is  always required. If your tape is unlabelled,
specify the same value as the visual-identifier.

 (visual-identifier|VID)

Specifies the contents of the sticky label on the volume reel itself.
VID's must be exactly 6 characters.
The visual-identifier must be a CERN tape number (5 digits followed
be a check letter), an experimental tape number, (1 or 2 characters
followed by 5 or 4 digits) or an XIN/XUT number.
XIN tapes are read only.

The visual-identifier parameter is  always required.

 logical-name[:]

Defines a 1- through  63-alphanumeric  character  string  logical
name to be associated with the staged data.
The logical name must be used to reference the staged tape data.

If you do not specify a logical name, the STAGE  command  assigns
the  default  logical  name  STAGE$volume-label.

The logical name is entered into the JOB logical name table, so that
```

```
STAGE may be called from a program using LIB$SPAWN. Once the sub-process
has completed, the parent process may then access the staged data via
this logical name.
3 /APPEND

This qualifier invokes output tape staging. Any existing data
is read from the corresponding tape file, before control is
returned to the user. This file must be opened by the user
with the APPEND attribute.

3 /BLOCKSIZE

Use the /BLOCKSIZE to indicate the blocksize of the tape file.
If not specified, STAGE uses a default of 32256 bytes, unless the tape is
labelled, in which case it reads the blocksize from the tape label.

3 /DIRECT

Use the /DIRECT qualifier if you wish to read or write directly to tape.
STAGE will perform the tape mount and file positioning.

3 /NAME=name

Use the /NAME qualifier to indicate the name of file that you wish
to stage. The filename need not be specified for unlabelled tapes. For
labelled tapes, the filename must match that recorded in the tape label.

3 /NUMBER=number

Use the /NUMBER qualifier to indicate the file sequence number that
you wish to stage. If not specified, the first file is staged. STAGE
understands tape labels and does not consider these as data files.

3 /FIXED

Use the /FIXED qualifier to request that the resultant disk file should
have fixed length records. If not specified, the format of the input tape
file is used, if the tape is labelled.


3 /GENERIC=device-type

   /GENERIC=TA90 (D)

 Use this qualifier to instruct STAGE to allocate a certain
 device type. If not specified, the generic device will default to
 TA90 - the IBM 3480 compatible drive.

 The generic devices available on your system are defined by the system
 manager as logical names. For example, the logical name for TA90 drives
 is SETUP_TA90S. The following table gives some examples.

 Generic Device type     Characteristics     Logical name
 ===================     ===============     ============
 TA90 (or 3480)          3480 cartridge      SETUP_TA90S (or SETUP_3480S)
 TA78                    6250 bpi 9 track    SETUP_TA78S
 TK50                    TK50 cartridge      SETUP_TK50S
```

```
 N.B. if /GENERIC is NOT specified and STAGE has been installed with
 the TMS option, STAGE will obtain the correct device information from
 the TMS.

3 /IN

Use this qualifier to request input tape staging. This qualifier is
the default stage option, if none of the other options (e.g. /WRITE,
 /LIST,/QUERY) is specified.

3 /KEEP

Use this qualifier in conjunction with the /WRITE qualifier to preserve
your data on disk after explicitly writing it to tape. Once the data
has been written to tape, the staged data is treated as if it had
just been input-staged and will eventually be deleted by the staging
system.

3 /NOPRESTAGE

This qualifier invokes output tape staging and is equivalent to a STAGE/APPEND,
except that no prestage of existing data from tape to disk is performed.
In  addition, STAGE will not check the output tape. It may only be
used after a STAGE/OUT command, and when some data has already
been written to disk.

3 /NREAD=n

Use the /NREAD qualifier to request how many tape records be read.
If not specified, 20 Megabytes will be staged. You cannot specify
both /NREAD and /SIZE.

3 /OUT

This qualifier invokes output tape staging.
If this qualifier is specified, no pre-stage of existing data
is performed. However, the output tape is mounted to ensure that
it exists and has the correct VOL1 label.


3 /RECORDSIZE=n

Use the /RECORDSIZE parameter to indicate the recordsize of the tape file.
If not specified, STAGE uses a default of 32256 bytes, unless the tape is
labelled, in which case it reads the recordsize from the tape label.

3 /REPLACE

Use the /REPLACE qualifier to request that existing staged data be replaced
by re-reading the input tape. Data is normally only re-staged if more data
is required, or if the disk file no longer exists.

3 /SIZE=n

Use the /SIZE qualifier to request that 'n' Megabytes be staged from tape.
If not specified, 20 MB will be staged.
```

If /SIZE=0 is specified, the complete tape file will be copied to disk.
If any data already resides on disk, the STAGE operation will not be
performed. To force STAGE to replace an existing file with a complete
new file, use /SIZE=0 with the /REPLACE qualifier.

3 /VARIABLE

Use the /VARIABLE qualifier to request that the output disk file have
variable length records. If not specified, STAGE will use the input record
format, if the tape is labelled.

3 /WRITE

This qualifier is valid only for output tape staging.
Once you have finished writing data to your file, you may request that
the data is immediately copied to tape by use of the /WRITE qualifier.
Use of the /WRITE qualifier is encouraged, to optimize use of the
staging disk. If STAGE is not reinvoked with the /WRITE qualifier,
the data will be moved to tape when disk space becomes low.

3 Examples

```
VXCRNB       stage/generic=rv20/name=test.bck jds001 jds001
%STAGE-I-TAPEINFO, file 1 from tape JDS001 (VSN JDS001), label type ASCII , file
%STAGE-I-FILENAME, corresponding disk filename is JDS001.1.ASCII.JDS001
%STAGE-I-DEFSIZE, 20 Megabytes will be staged by default
%STAGE-I-FILEFND, file already on disk, size =  17892 blocks
%STAGE-I-LARGER, more data is requested that currently resides on disk
%STAGE-I-STGREQ, a stage operation is required
%SETUP-I-PENDING, your tape request is being processed, please wait
%SETUP-I-ALLOC, _VXCRNB$MUA0: allocated
%SETUP-I-ASCII, tape JDS001 contains labels encoded in ASCII
%STAGE-I-STAGING, starting stage of data from tape to disk
%STAGE-I-EOF, EOF on input after    9160704 bytes
%STAGE-I-LOGNAME, logical name for your staged file is STAGE$JDS001
```

2 Input_Staging

Use the STAGE command to initiate input tape staging of a single tape file.
As input staging is the default, you do not need to specify any qualifiers
to indicate that input staging is required. Note that, if the requested
file is currently being output staged, STAGE will not permit you access.

STAGE uses SETUP routines if a tape mount is required. See HELP SETUP for
details of the SETUP command.

2 Output_staging

Output staging may be requested by use of the /APPEND or /OUT qualifiers.
It is otherwise identical to the input STAGE command.
See the section Command_Syntax for full details on qualifiers.

The /APPEND and /OUT qualifiers will cause STAGE to check that your
output tape exists, and then create a file of the desired size and format.
STAGE assigns a logical name to the disk file, by which you should reference
the file.

```
If the file already exists on tape, STAGE is able to obtain the format
from the tape label. If not, you must specify the format with the
appropriate qualifiers.

2 Managing_staged_data

The preferred method of managing your staged data is through the STAGE/LIST
command. This permits you to perform other STAGE functions, such as QUERY
or CLEAR commands, without having to specify the full name of the staged
data file.

2 /CLEAR tape(s)

  Use the /CLEAR qualifier to delete existing staged data when
it is no longer required. You may only clear data that has been
staged by your UIC group. See HELP STAGE NAMING_CONVENTION for
details of how to specify the name of the tapes to be cleared.

See also HELP STAGE/LIST for a more user-friendly interface to
STAGE/CLEAR.

An alternative method of delete STAGEd data from DCL, or from within
a program is to use the logical name assigned to your data.

3 Examples

The following example shows how data may be deleted using the logical
name assigned by STAGE.

$STAGE/NAME=TEST.DAT XQ1170 XIN130 ! Stage data
... Process data ...
$!Finished with this data - delete it
$DELETE STAGE$XIN130


3 Parameters

4 Tape(s)

The only parameter that is normally required for the /CLEAR function
is the tape, or list of tapes, that you wish to clear.
See HELP STAGE NAMING_CONVENTION for details of how to specify
the name of the tapes to be cleared.

4 /OUT

If the tape that you wish to clear is currently being staged out,
you must specify /OUT. This will clear the file without writing
any data to tape.

2 /LIST

Use this qualifier by itself to invoke a FILELIST like interface to help
you manage your staged data.

2 Naming_Convention
```

```
 A tape is uniquely identified by is magnetic label (VSN),
 label on the tape reel (VID), filenumber and label type. For the
 /CLEAR subcommand, you must use the format VSN_VID.FILENUMBER_LABELTYPE
 to indicate which tapes are to be cleared. Trailing items are defaulted,
 thus JDS001 defaults to JDS001_JDS001.1_ASCII

3 Examples

To indicate a tape with VSN NC0383, VID XIN127, filenumber 3 and labeltype
of EBCDIC, use the format NC0383_XIN127.3_EBCDIC

2 /QUERY

Use the /QUERY qualifier to find out if a given group is enabled for
tape staging, to list which groups are enabled, or query the status
of one or more tapes.

See also HELP STAGE/LIST for a more user-friendly interface to
STAGE/QUERY.

3 /GROUP=list

Use the /GROUP qualifier to indicate which UIC groups you wish to
query. STAGE will indicate whether each of the groups listed may or
may not using the STAGING facility, and which tapes are staged for
each group. The name of each tape is followed by a /I or /O to indicate
whether the tape is being staged in or out.

3 /GROUPS

Use the /GROUPS qualifier to indicate which UIC groups are enabled
for tape staging.

3 /LIST

Use the /LIST qualifier to invoke a FILELIST interface to manage
your staged tape files.

3 /TAPES=list

Use the /TAPES qualifier to query the status of one or more tapes.
STAGE will indicate whether the tapes have been staged, or are in
the process of being staged. If the tapes are currently being staged,
more detailed information, such as 'waiting for tape mount', is also
displayed. The name of each tape is followed by a /I or /O to indicate
whether the tape is being staged in or out.

2 Calling STAGE from FORTRAN

STAGE may be called from FORTRAN, or any other language, using
the VAX/VMS Run-time library routines. For example, a program
may wish to input stage some data, process it, and delete the file.
This it could be by

a) Call LIB$SPAWN to issue the STAGE command, specifying that the
parent process hibernates until the subprocess completes.
```

```
b) Once the subprocess has completed, the data will have been staged to
disk, unless a error code is returned.

c) The program can then continue to process this data.

d) Once processing has completed, it can delete the staged file using
LIB$DELETE_FILE, using the logical name returned by STAGE.

      integer status

      status = lib$spawn('$STAGE/NAME=TEST.DAT XQ1170 XIN132 /SIZE=10')
      if (.not. status) call lib$signal(%val(status))

*
*      Process data
*

*
*      Delete file
*
      status = lib$delete_file('STAGE$XIN132')
      if (.not. status) call lib$signal(%val(status))

      end
```

## 2.5 TAPECOPY

```
1 TAPECOPY

Use this command to initiate a complete tape to tape copy,
including tape labels, if any.

This command takes no qualifiers - all required information
is prompted for.

As well as copying one tape to another, this command can be
used to copy from 6250 bpi tapes to 3480 cartridges and so
on, by specifying the correct input and output device types.

2 Examples

The example below shows how to copy an input tape to an output 3480
cartridge. The job is executed in batch.


VXCRNA? tapecopy

Command file to copy an entire input tape (including tape labels, if any)
to an output tape.

As well as performing a tape to tape copy,  this command file allows you
```

```
to copy a 6250 bpi tape to a 3480 cartridge etc, by correctly specifying
the input and output devices required.

If you do not know the tape labels of either input or output tapes, use the
LABELDUMP command to find them, prior to running this command file.

VID of input tape       xin103
VSN of input tape       (CR=unlabelled) xy0315
Labeltype of input tape       (ASCII/EBCDIC/NONE) ascii

Valid devices are:

TA78 (for normal 6250/1600 bpi tapes)
A480 (for IBM 3480-style cartridges)

Input device type       (Return=TA78)
VID of output tape       cut215
VSN of output tape       (CR=unlabelled)
Output device type       (Return=TA78) a480
Run in BATCH       (Return=NO) y
Job COPY_XIN103_TO_CUT215 (queue CLU$TAPES, entry 43) started on VXCRNA_SYS$TAPE
```

## 2.6 WRTAPE

```
1 WRTAPE

  Use the WRTAPE command to copy a disk file to tape, writing EBCDIC
  or ASCII tape labels.

  WRTAPE is essentially the same as STAGE/OUT except that it allows any
  existing file to be specified, using the /INFILE qualifier.

  For all other qualifiers, see the description of the STAGE /OUTPUT command.

2 Examples

WRTAPE/INFILE=MYDATA.DAT JDS001 CUT123 /GENERIC=TA90 /NAME=DELPHI/NUMBER=3

The above command writes file MYDATA.DAT to file 3 of the tape with VID CUT123.
The file name to be written on the tape is DELPHI.
```

## 2.7 XTAPE

```
1 XTAPE

  Use the XTAPE command to examine the labels of a tape and optionally
  dump blocks from the specified number of files.
```

```
   Format:

   XTAPE [(visual identifier)]

2 Parameters
  (visual identifier)

  Specifies the number of the tape on the tape reel itself.
  If you do not specify this parameter, you are prompted for it.

2 Qualifiers
/BLOCKS

   /BLOCKS=1 (D)

  Use this qualifier to specify how many blocks from each file are to be
  dumped.

/FILES

   /FILES=1 (D)

 Use this qualifier to specify how many files are to be dumped.

/GENERIC

   /GENERIC=TA90 (D)

 N.B. if XDUMP has been installed with the TMS option,
 as is the case on VXCERN, XDUMP will obtain the correct
 device information from the TMS and /GENERIC should NOT be specified.

 Use this qualifier to instruct XDUMP to allocate a certain
 device type. If not specified, the generic device will default to
 TA90 - the IBM 3480 compatible drive.

 The generic devices available on your system are defined by the system
 manager as logical names, e.g. SETUP_TA90S, SETUP_TA78S.

/DEBUG

 This qualifer requires SYSPRV and is primarily of use for debugging
 new versions of the XTAPE command. If specified, extra debug information
 will be displayed during command processing.

/WAIT

   /WAIT     (Default in BATCH)
   /NOWAIT   (Default otherwise)

  This parameter controls whether LABELDUMP will exit if all tape drives
  are currently in use, or wait until one becomes free. If executing
  in BATCH mode, /WAIT is in effect unless explicitly negated.
```

# Chapter 3: VAXTAP help files for systems with a TMS interface

The following help files are for systems with a TMS interface.

## 3.1 EINIT

```
1 EINIT

  Use the EINIT command to write a magnetic tape label encoded in EBCDIC.

  Format:

  RUN SETUP_EXE:EINIT

  The volume to be labelled must already be mounted /FOREIGN/WRITE on
  the tape unit with logical name INTAPE.

  The utility will prompt for the volume label to be written.
```

## 3.2 LABELDUMP

```
1 LABELDUMP


   Use the LABELDUMP command to find out the magnetically recorded label
   (if any) of a magnetic tape.

   Format:

   LABELDUMP [(visual identifier)]

   The volume label, as defined in the Tape Management System (TMS)
   may be obtained by typing:

   SYSREQ QUERY VID vid (VSN



2 Parameters
  (visual identifier)

  Specifies the number of the tape on the tape reel itself.
  If you do not specify this parameter, you are prompted for it.

2 Qualifiers

/DEBUG

 This qualifer requires SYSPRV and is primarily of use for debugging
 new versions of the LABELDUMP command. If specified, extra debug information
 will be displayed during command processing.

/WAIT

   /WAIT     (Default in BATCH)
   /NOWAIT   (Default otherwise)
```

```
  This parameter controls whether LABELDUMP will exit if all tape drives
  are currently in use, or wait until one becomes free. If executing
  in BATCH mode, /WAIT is in effect unless explicitly negated.


2 Examples

  LABELDUMP XIN101

  Dumps the label of XIN tape XIN101.

  SYSREQ TMS Q VID I29021 (VSN

  Display the VSN as defined in the TMS
```

## 3.3 SETUP

```
1 SETUP

 The SETUP command should always be used to mount magnetic tapes,
 rather than the standard VMS MOUNT command.
 SETUP command will ALLOCATE a tape drive, MOUNT the tape
 and perform volume-label checking. The tape is then re-mounted via a
 standard VMS mount command and made available for user-processing.

 In addition to the qualifiers presented below, any MOUNT qualifers
 may be used. See HELP MOUNT for further information on additional
 qualifiers.

 N.B. SETUP has been installed with the TMS option on this node.
 The correct device and label information from the TMS and the
 qualifiers /GENERIC, /LABEL should NOT be specified.

 Format:

      SETUP [(volume-label|VSN)] [(visual-identifier|VID)] [(logical-name)]

2 Parameters
 (volume-label|VSN)

  Specifies the MAGNETICALLY recorded label on the volume.
  Labels can have from 1 through 6 characters. This is the
  equivalent of the IBM MVS volume serial name (VSN) on the DD statement.

  The volume-label parameter is  always required. If your tape is unlabelled,
  specify the same value as the visual-identifier.

  N.B. if the VSN is in lower or mixed case, it must be enclosed in
  double quotes.

 (visual-identifier|VID)

  Specifies the contents of the sticky label on the volume reel itself.
  VID's must be exactly 6 characters.
  The visual-identifier must be a CERN tape number (5 digits followed
```

```
  be a check letter), an experimental tape number, (1 or 2 characters
  followed by 5 or 4 digits) or an XIN/XUT number.
  XIN tapes are read only.

  The visual-identifier parameter is  always required.

 logical-name[:]

  Defines a 1- through  63-alphanumeric  character  string  logical
  name to be associated with the device allocated.
  The logical name must be used to reference the drive on which the
  tape is mounted.

  If you do not specify a logical name, the SETUP  command  assigns
  the  default  logical  name  TAPE$volume-label.

2 Qualifiers
/BACKUP

 If you specify /BACKUP, your tape will be mounted /FOREIGN, after
 SETUP has verified that the correct volume-label has been specified.
 Use this qualifier to read and write tapes in BACKUP format, in
 addition to the relevant /LABEL qualifier.

/DEBUG

 This qualifer requires SYSPRV and is primarily of use for debugging
 new versions of the SETUP command. If specified, extra debug information
 will be displayed during command processing.

/FOREIGN

 Use of this qualifier requires VOLPRO or BYPASS privilege.
 Without privilege, tapes may only be mounted /FOREIGN if the
 /BACKUP qualifier is specified, together with the correct
 /LABEL qualifier and value. See the description of the /BACKUP
 and /LABEL qualifiers and the examples section for more details.

 If you do not know the magnetic label of a tape, use the LABELDUMP
 command to determine it.
/WAIT

   /WAIT     (Default in BATCH)
   /NOWAIT   (Default otherwise)

  This parameter controls whether SETUP will exit if all tape drives
  are currently in use, or wait until one becomes free. If executing
  in BATCH mode, /WAIT is in effect unless explicitly negated.

/WRITE

   /WRITE
   /NOWRITE  (D)

 Controls   whether  the  volume  can  be  written.   /NOWRITE  is
 equivalent to write-locking the device.
```

```
2 Mounting_FOREIGN_tapes

On the CERN Computer Centre VAXcluster,tapes do not normally need
to be mounted /FOREIGN. The exception is for processing with BACKUP.

To mount a tape for processing by BACKUP (or any other program that
requires the tape to be mounted /FOREIGN), use

SETUP/BACKUP.

Note that you MUST specify the correct volume-label, if any.


2 Examples
Setup a tape for read only.
$ .....
$ SETUP MX1015 MX1015            ! If the SETUP fails, the job will exit.
$ ASSIGN TAPE$MX1015 FOR001
$ RUN DSTANAL
$ EXIT

Setup a tape for write

$ ON ERROR THEN GOTO ABEND       ! Protect against problems with SETUP etc.
$ SETUP/NOWRITE/WRITE 939927 XIN112 MYDRIVE
$ ASSIGN MYDRIVE:  FOR001        ! Processing of data sets is up to user.
$ RUN PROG
$ EXIT
$ !
$ABEND:
$ WRITE SYS$OUTPUT "Program abended"
$ EXIT

Mount a labelled tape for writing to with BACKUP.

$SETUP/WRITE/BACKUP MICRO PR9902

Mount a tape written on a data acquisition VAX
blocksize = 3600 bytes and recordsize = 3600 bytes.

$SETUP/BLOCK=3600/RECORD=3600 MC1009 MC1009
$! Skip past volume label to logical file 1
$SET MAGTAPE/SKIP=FILES:1 TAPE$MC1009
$RUN MYPROG ! Uses EPIO, reading from unit 1.
```

## 3.4 STAGE

```
1 STAGE

Use the STAGE command to perform input or output staging of tape
data, clear existing staged data or query the status of staged
data.

Each group must be explicitly enabled to use tape staging. If you
find that your group is not enabled, please contact the UCO.

2 Command_Syntax
```

The syntax of the STAGE command is essentially that of SETUP, with some extensions. Only these extensions are described below.

The format of the STAGE command is:

```
STAGE [(volume-label|VSN)] [(visual-identifier|VID)] [(logical-name)] -
  /NAME=fn/NUMBER=n
```

3 Parameters
 (volume-label|VSN)

Specifies the MAGNETICALLY recorded label on the volume.
Labels can have from 1 through 6 characters. This is the
equivalent of the IBM MVS volume serial name (VSN) on the DD statement.

The volume-label parameter is always required. If your tape is unlabelled, specify the same value as the visual-identifier.

 (visual-identifier|VID)

Specifies the contents of the sticky label on the volume reel itself.
VID's must be exactly 6 characters.
The visual-identifier must be a CERN tape number (5 digits followed
be a check letter), an experimental tape number, (1 or 2 characters
followed by 5 or 4 digits) or an XIN/XUT number.
XIN tapes are read only.

The visual-identifier parameter is always required.

 logical-name[:]

Defines a 1- through 63-alphanumeric character string logical
name to be associated with the staged data.
The logical name must be used to reference the staged tape data.

If you do not specify a logical name, the STAGE command assigns
the default logical name STAGE$volume-label.

The logical name is entered into the JOB logical name table, so that
STAGE may be called from a program using LIB$SPAWN. Once the sub-process
has completed, the parent process may then access the staged data via
this logical name.
3 /APPEND

This qualifier invokes output tape staging. Any existing data
is read from the corresponding tape file, before control is
returned to the user. This file must be opened by the user
with the APPEND attribute.

3 /BLOCKSIZE

Use the /BLOCKSIZE to indicate the blocksize of the tape file.
If not specified, STAGE uses a default of 32256 bytes, unless the tape is
labelled, in which case it reads the blocksize from the tape label.

3 /DIRECT

```
Use the /DIRECT qualifier if you wish to read or write directly to tape.
STAGE will perform the tape mount and file positioning.

3 /NAME=name

Use the /NAME qualifier to indicate the name of file that you wish
to stage. The filename need not be specified for unlabelled tapes. For
labelled tapes, the filename must match that recorded in the tape label.

3 /NUMBER=number

Use the /NUMBER qualifier to indicate the file sequence number that
you wish to stage. If not specified, the first file is staged. STAGE
understands tape labels and does not consider these as data files.

3 /FIXED

Use the /FIXED qualifier to request that the resultant disk file should
have fixed length records. If not specified, the format of the input tape
file is used, if the tape is labelled.

3 /IN

Use this qualifier to request input tape staging. This qualifier is
the default stage option, if none of the other options (e.g. /WRITE,
 /LIST,/QUERY) is specified.

3 /KEEP

Use this qualifier in conjunction with the /WRITE qualifier to preserve
your data on disk after explicitly writing it to tape. Once the data
has been written to tape, the staged data is treated as if it had
just been input-staged and will eventually be deleted by the staging
system.

3 /NOPRESTAGE

This qualifier invokes output tape staging and is equivalent to a STAGE/APPEND,
except that no prestage of existing data from tape to disk is performed.
In  addition, STAGE will not check the output tape. It may only be
used after a STAGE/OUT command, and when some data has already
been written to disk.

3 /NREAD=n

Use the /NREAD qualifier to request how many tape records be read.
If not specified, 20 Megabytes will be staged. You cannot specify
both /NREAD and /SIZE.

3 /OUT

This qualifier invokes output tape staging.
If this qualifier is specified, no pre-stage of existing data
is performed. However, the output tape is mounted to ensure that
it exists and has the correct VOL1 label.
```

```
3 /RECORDSIZE=n

Use the /RECORDSIZE parameter to indicate the recordsize of the tape file.
If not specified, STAGE uses a default of 32256 bytes, unless the tape is
labelled, in which case it reads the recordsize from the tape label.

3 /REPLACE

Use the /REPLACE qualifier to request that existing staged data be replaced
by re-reading the input tape. Data is normally only re-staged if more data
is required, or if the disk file no longer exists.

3 /SIZE=n

Use the /SIZE qualifier to request that 'n' Megabytes be staged from tape.
If not specified, 20 MB will be staged.

If /SIZE=0 is specified, the complete tape file will be copied to disk.
If any data already resides on disk, the STAGE operation will not be
performed. To force STAGE to replace an existing file with a complete
new file, use /SIZE=0 with the /REPLACE qualifier.

3 /VARIABLE

Use the /VARIABLE qualifier to request that the output disk file have
variable length records. If not specified, STAGE will use the input record
format, if the tape is labelled.

3 /WRITE

This qualifier is valid only for output tape staging.
Once you have finished writing data to your file, you may request that
the data is immediately copied to tape by use of the /WRITE qualifier.
Use of the /WRITE qualifier is encouraged, to optimize use of the
staging disk. If STAGE is not reinvoked with the /WRITE qualifier,
the data will be moved to tape when disk space becomes low.

3 Examples

VXCRNB       stage/generic=rv20/name=test.bck jds001 jds001
%STAGE-I-TAPEINFO, file 1 from tape JDS001 (VSN JDS001), label type ASCII , file
%STAGE-I-FILENAME, corresponding disk filename is JDS001.1.ASCII.JDS001
%STAGE-I-DEFSIZE, 20 Megabytes will be staged by default
%STAGE-I-FILEFND, file already on disk, size =  17892 blocks
%STAGE-I-LARGER, more data is requested that currently resides on disk
%STAGE-I-STGREQ, a stage operation is required
%SETUP-I-PENDING, your tape request is being processed, please wait
%SETUP-I-ALLOC, _VXCRNB$MUA0: allocated
%SETUP-I-ASCII, tape JDS001 contains labels encoded in ASCII
%STAGE-I-STAGING, starting stage of data from tape to disk
%STAGE-I-EOF, EOF on input after    9160704 bytes
%STAGE-I-LOGNAME, logical name for your staged file is STAGE$JDS001

2 Input_Staging

Use the STAGE command to initiate input tape staging of a single tape file.
```

```
As input staging is the default, you do not need to specify any qualifiers
to indicate that input staging is required. Note that, if the requested
file is currently being output staged, STAGE will not permit you access.

STAGE uses SETUP routines if a tape mount is required. See HELP SETUP for
details of the SETUP command.

2 Output_staging

Output staging may be requested by use of the /APPEND or /OUT qualifiers.
It is otherwise identical to the input STAGE command.
See the section Command_Syntax for full details on qualifiers.

The /APPEND and /OUT qualifiers will cause STAGE to check that your
output tape exists, and then create a file of the desired size and format.
STAGE assigns a logical name to the disk file, by which you should reference
the file.

If the file already exists on tape, STAGE is able to obtain the format
from the tape label. If not, you must specify the format with the
appropriate qualifiers.

2 Managing_staged_data

The preferred method of managing your staged data is through the STAGE/LIST
command. This permits you to perform other STAGE functions, such as QUERY
or CLEAR commands, without having to specify the full name of the staged
data file.

2 /CLEAR tape(s)

  Use the /CLEAR qualifier to delete existing staged data when
it is no longer required. You may only clear data that has been
staged by your UIC group. See HELP STAGE NAMING_CONVENTION for
details of how to specify the name of the tapes to be cleared.

See also HELP STAGE/LIST for a more user-friendly interface to
STAGE/CLEAR.

An alternative method of delete STAGEd data from DCL, or from within
a program is to use the logical name assigned to your data.

3 Examples

The following example shows how data may be deleted using the logical
name assigned by STAGE.

$STAGE/NAME=TEST.DAT XQ1170 XIN130 ! Stage data
... Process data ...
$!Finished with this data - delete it
$DELETE STAGE$XIN130


3 Parameters

4 Tape(s)
```

```
The only parameter that is normally required for the /CLEAR function
is the tape, or list of tapes, that you wish to clear.
See HELP STAGE NAMING_CONVENTION for details of how to specify
the name of the tapes to be cleared.

4 /OUT

If the tape that you wish to clear is currently being staged out,
you must specify /OUT. This will clear the file without writing
any data to tape.

2 /LIST

Use this qualifier by itself to invoke a FILELIST like interface to help
you manage your staged data.

2 Naming_Convention

 A tape is uniquely identified by is magnetic label (VSN),
 label on the tape reel (VID), filenumber and label type. For the
 /CLEAR subcommand, you must use the format VSN_VID.FILENUMBER_LABELTYPE
 to indicate which tapes are to be cleared. Trailing items are defaulted,
 thus JDS001 defaults to JDS001_JDS001.1_ASCII

3 Examples

To indicate a tape with VSN NC0383, VID XIN127, filenumber 3 and labeltype
of EBCDIC, use the format NC0383_XIN127.3_EBCDIC

2 /QUERY

Use the /QUERY qualifier to find out if a given group is enabled for
tape staging, to list which groups are enabled, or query the status
of one or more tapes.

See also HELP STAGE/LIST for a more user-friendly interface to
STAGE/QUERY.

3 /GROUP=list

Use the /GROUP qualifier to indicate which UIC groups you wish to
query. STAGE will indicate whether each of the groups listed may or
may not using the STAGING facility, and which tapes are staged for
each group. The name of each tape is followed by a /I or /O to indicate
whether the tape is being staged in or out.

3 /GROUPS

Use the /GROUPS qualifier to indicate which UIC groups are enabled
for tape staging.

3 /LIST

Use the /LIST qualifier to invoke a FILELIST interface to manage
your staged tape files.

3 /TAPES=list
```

```
Use the /TAPES qualifier to query the status of one or more tapes.
STAGE will indicate whether the tapes have been staged, or are in
the process of being staged. If the tapes are currently being staged,
more detailed information, such as 'waiting for tape mount', is also
displayed. The name of each tape is followed by a /I or /O to indicate
whether the tape is being staged in or out.

2 Calling STAGE from FORTRAN

STAGE may be called from FORTRAN, or any other language, using
the VAX/VMS Run-time library routines. For example, a program
may wish to input stage some data, process it, and delete the file.
This it could be by

a) Call LIB$SPAWN to issue the STAGE command, specifying that the
parent process hibernates until the subprocess completes.

b) Once the subprocess has completed, the data will have been staged to
disk, unless a error code is returned.

c) The program can then continue to process this data.

d) Once processing has completed, it can delete the staged file using
LIB$DELETE_FILE, using the logical name returned by STAGE.

      integer status

      status = lib$spawn('$STAGE/NAME=TEST.DAT XQ1170 XIN132 /SIZE=10')
      if (.not. status) call lib$signal(%val(status))

*
*      Process data
*


*
*      Delete file
*
      status = lib$delete_file('STAGE$XIN132')
      if (.not. status) call lib$signal(%val(status))

      end
```

## 3.5 TAPECOPY

```
1 TAPECOPY

Use this command to initiate a complete tape to tape copy,
including tape labels, if any.

This command takes no qualifiers - all required information
is prompted for or obtained from the Tape Management System (TMS).

As well as copying one tape to another, this command can be
used to copy from 6250 bpi tapes to 3480 cartridges and so
on, by specifying the correct input and output device types.
```

```
2 Examples

VXCRNA? tapecopy

Command file to copy an entire input tape (including tape labels, if any)
to an output tape.

All tape information, apart from input and output VIDs, is obtained
from the Tape Management System (TMS).

VID of input tape       i29021
Obtaining volume information from TMS...
Input VID   : I29021
Input VSN   : I29021
Input label : EBCDIC
Input device: SMCF
VID of output tape       i29022
Obtaining volume information from TMS...
Output VID   : I29022
Output VSN   : I29022
Output label : EBCDIC
Output device: SMCF
Run in BATCH       (Return=NO)yes
Job COPY_I29021_TO_I29022 (queue CLU$TAPES, entry 897) pending
     pending status caused by queue busy
```

## 3.6 WRTAPE

```
1 WRTAPE

  Use the WRTAPE command to copy a disk file to tape, writing EBCDIC
  or ASCII tape labels, depending on the currently labeltype as defined
  in the Tape Management System (TMS). Type

  SYSREQ TMS QVOL vid

  to display the current labeltype (SL=labels encoded in EBCDIC, (IBM format)
                                    AL=labels encoded in ASCII,  (VAX format)
                                    NL=no labels).

  WRTAPE is essentially the same as STAGE/OUT except that it allows any
  existing file to be specified, using the /INFILE qualifier.

  For all other qualifiers, see the description of the STAGE /OUTPUT command.

2 Examples

WRTAPE/INFILE=MYDATA.DAT JDS001 CUT123 /NAME=DELPHI/NUMBER=3

The above command writes file MYDATA.DAT to file 3 of the tape with VID CUT123.
The file name to be written on the tape is DELPHI.
```

## 3.7 XTAPE

```
1 XTAPE
```

```
  Use the XTAPE command to examine the labels of a tape and optionally
  dump blocks from the specified number of files.


   Format:

   XTAPE [(visual identifier)]

2 Parameters
  (visual identifier)

  Specifies the number of the tape on the tape reel itself.
  If you do not specify this parameter, you are prompted for it.

2 Qualifiers
/BLOCKS

   /BLOCKS=1 (D)

  Use this qualifier to specify how many blocks from each file are to be
  dumped.

/FILES

   /FILES=1 (D)

 Use this qualifier to specify how many files are to be dumped.

/GENERIC

   /GENERIC=TA90 (D)

 N.B. if XDUMP has been installed with the TMS option,
 as is the case on VXCERN, XDUMP will obtain the correct
 device information from the TMS and /GENERIC should NOT be specified.

 Use this qualifier to instruct XDUMP to allocate a certain
 device type. If not specified, the generic device will default to
 TA90 - the IBM 3480 compatible drive.

 The generic devices available on your system are defined by the system
 manager as logical names, e.g. SETUP_TA90S, SETUP_TA78S.

/DEBUG

 This qualifer requires SYSPRV and is primarily of use for debugging
 new versions of the XTAPE command. If specified, extra debug information
 will be displayed during command processing.

/WAIT

   /WAIT     (Default in BATCH)
   /NOWAIT   (Default otherwise)

  This parameter controls whether LABELDUMP will exit if all tape drives
  are currently in use, or wait until one becomes free. If executing
```

in BATCH mode, /WAIT is in effect unless explicitly negated.

# Part III

# VAXTAP – Installation and Management Guide

# Chapter 4: Installing VAXTAP

The following steps are required to install VAXTAP.

1 Obtain the VAXTAP.CAR, e.g. copy from VXCRNA, ASIS, etc.
2 Extract the installation command file as described below
3 Run the installation command file to generate the executables, help files and command files
4 Define the appropriate logical names

## 4.1 Extracting the installation command file

The package is installed by running a command file that can be generated by the following PATCHY run:

```
$YPATCHY CERN:[PRO.PAM]VAXTAP.PAM INSTALL.COM :GO
+USE,INSTALL,T=EXE.
+PAM.
+QUIT
```

## 4.2 Running the installation command file

Once this command file has been extracted, installation of the package proceeds by typing

```
@INSTALL
```

and answering the questions. If VAXTAP is to be installed on a system without access to the HEPVM Tape Management System, as is likely to be the case when installing it outside CERN, answer NO to the first question. Answering A or ALL to the second question will cause the installation to complete without any further dialogue.

This procedure generates the executables, help files and command files that are required. An example session is shown below.

**Installation of VAXTAP**

```
VXCRNA? @install

You may install the VAXTAP utilities with or without the TMS option.
If you chose the TMS option, information such as the generic device
type, label type and checks on the volumes existance will be performed
by the TMS. At CERN, this option requires:

- Wollongong TCP/IP
- A SYSREQ interface

Please contact J. Shiers for more information

Use TMS flag       (Y/N/default=N) y
LIB$ = CERN:[PRO.LIB]PACKLIB/LIB,KERNLIB/LIB,SYS$LIBRARY:VAXCRTL/LIB
>>> Step 1: extract source files from PAM

(PATCHY output suppressed)
```

```
>>> Step 2: build VAXTAP library
>>> Step 3: create command files
Creating SETUP_STARTUP.COM...
Creating CLEAN_STAGE.COM...
Creating STAGELIST.COM...
Creating TAPECOPY.COM...
Creating TPDAEMON.COM...
Creating STAGECLN.COM...
Creating DUMMY.COM...
Creating ENABLE_STAGING.COM...
>>> Step 4: build .EXE files
Build EINIT       (Y/N/A/Q) a
>>> Building EINIT...
Patchy step...
Compile step...
Compiling CLD file...
Link step...
Multinet TCP/IP...
>>> Building LABELDUMP...
Patchy step...
Compile step...
Compiling CLD file...
Link step...
Multinet TCP/IP...
>>> Building SETUP...
Patchy step...
Compile step...
Compiling CLD file...
Link step...
Multinet TCP/IP...
>>> Building STAGE...
Patchy step...
Compile step...
Compiling CLD file...
Link step...
Multinet TCP/IP...
>>> Building STAGECLN...
Patchy step...
Compile step...
Compiling CLD file...
Link step...
Multinet TCP/IP...
>>> Building TAPECOPY...
Patchy step...
Compile step...
Compiling CLD file...
Link step...
Multinet TCP/IP...
>>> Building WRTAPE...
Compile step...
Compiling CLD file...
Link step...
Multinet TCP/IP...
>>> Building XTAPE...
Patchy step...
Compile step...
```

```
Compiling CLD file...
Link step...
Multinet TCP/IP...

You should now do the following:
 1) Modify SETUP_STARTUP.COM to suit your system
 2) Add the help files to the required help library
 3) Copy the .EXE files to the directory SETUP_EXE

See the file VAXTAP.SGML for details.
```

## 4.3 Defining the logical names

The installation procedure generates a file SETUP_STARTUP.COM, which should be tailored to suit your system and then run at boot time.

The following logical name definitions are required in the LNM$SYSTEM table.

| | |
|---|---|
| SETUP_EXE | Points to the directory where the .EXE files are kept |
| SETUP_LOGS | Points to the directory for log files from cleanup jobs |
| SETUP_ENABLED | Defined as INTERACTIVE, if interactive tape mounting is to be allowed |
| SETUP_GENERIC | List of users for whom the check on generic device name against the TMS is not performed. |
| SETUP_MAXDEV | Maximum number of devices of a given type that can be allocated per user per node. If not defined, the default is 999. |
| SETUP_NOTAPES | Can be used to define a list of users that are not allowed to use the VAXTAP commands |
| SETUP_QUEUES | Points to a list of batch queue names in which SETUP commands may be issued (or * for no restriction) |
| STAGE_TAPES | Defined as YES if tape staging is enabled. |
| DISK$STAGE | Points to the disk (or volume set, stripe set or search list) to be used for tape staging. |

In addition, a logical name must exist for each generic device type. The logical name is SETUP_generic-deviceS, thus, for a generic device of type 8MM, the corresponding logical name is SETUP_8MMS. The generic device type is selected using the /GENERIC qualifier of the various VAXTAP commands.

**SETUP_STARTUP.COM**

```
$ !-------------------------------------------------------------------------*
$ !
$ !      Startup command file for SETUP/STAGE/LABELDUMP
$ !      Modify logical name definitions as required for your node.
$ !-------------------------------------------------------------------------*
$ !
$ !
$ !*** Following lines are CERN specific ***
$ !      Create lnm table for SETUP information ...
```

```
$ !      This is used at CERN in conjunction with the TMS
$        create/name_table/parent=lnm$system_table/prot=w:wred lnm$setup
$ !
$ !      Issue TMS flush command
$ !
$        machine = f$trnlnm("SYS$NODE") - "::"
$        command = "TMS FLUSH ALL SYSTEM ''machine'"
$        'command'
$ !*** End of CERN specific code ***
$ !
$ !      Define directory for .EXE files
$        define/system setup_exe cern_root:[exe]
$ !
$ !      Define symbols for commands
$ !
$        einit    :==$setup_exe:einit
$        labeldump:==$setup_exe:labeldump
$        setup    :==$setup_exe:setup
$        stage    :==$setup_exe:stage
$        wrtape   :==$setup_exe:wrtape
$        tapecopy :==@setup_exe:tapecopy ! Tapecopy is a command file
$        xtape    :==$setup_exe:xtape
$ !
$ !      Allow usage of tapes interactively
$ !
$        define/system setup_enabled "INTERACTIVE"
$ !
$ !      Disallow specific users from using tapes (useful to stop troublemakers)
$ !
$ !      define/system setup_notapes "DECNET,CERNET"
$ !
$ !      Allow tapes in these batch queues
$ !
$ !      define/system setup_queues "SYS$TAPES"
$ !      define/system setup_queues "SYS$BATCH,SYS$TAPES"
$ !      define/system setup_queues "*" ! all queues
$ !
$ !      Set up lists of available device types
$ !
$        define/system setup_tk50s "VSDD18$MKA700:"
$        define/system setup_8200s "UXDDB1$MUB0:"
$        define/system setup_exabytes setup_8200s ! Can also have aliases...
$ !
$ !      Allow tape staging
$ !
$        define/system stage_tapes "YES"
$ !      Must also ensure that DISK$STAGE exists...
$ !
$ !     Define directory for log files (CLEAN_STAGE.COM called from
$ !     STAGECLN)
$       if f$trnlnm("SYS$LOG") .eqs. ""
$          then
$             define/system setup_logs sys$manager
$          else
$             define/system setup_logs sys$log
$          endif
```

## 4.4 Installing the VAXTAP images

The VAXTAP-TMS interface consists of two parts. The first part, selected by the PATCHY flag TMS, obtains information such as the label type and device type from the TMS. If this option is selected, then the /LABEL and /GENERIC qualifiers should not be specified. If the are specified, the information must match that obtained from the TMS. This can be overridden for users in the list defined by the logical name SETUP_GENERIC.

In addition, the flag TMSMOUNT selects code which will record mounts in the TMS. If you have selected the TMSMOUNT option, then all images, with the exception of EINIT, must be installed with **SYSPRV** privilege. This is to permit the programs to communicate correctly with the TMS system.

An example of such an installation is shown below.

**Installing the VAXTAP images on a system with a TMS interface**

```
$ ins:==$install/command
$ INS
CREATE SETUP_EXE:SETUP/OPEN/SHARE/HEAD/PRIV=SYSPRV
```

In addition, the STAGE executable must be installed with SYSLCK privilege. This is required whether a TMS interface exists or not. SYSLCK is used to communicate with the process that cleans up the staging disk.

## 4.5 Configuring the staging system

To use the STAGE command, one must also allocate some disk space. It is recommended that a seperate disk or disks be used for staging, rather than mixing the staging area with, say, user disks. In fact, the current clean up procedures assume that this has been done and will **DELETE** and files which do not conform to the staging file name convention.

At CERN, the cluster size for the staging disk, which is in fact a stripe set, is set to 250 blocks. The average size of a staged file is typically 180 MB and so the default cluster size is far too low.

The staging disk may be also be a volume set or a search list. In the latter case, the staging system uses that disk with the most free blocks.

Once the staging disk has been allocated, access must be enabled on a group basis. This is typically by UIC group, such as PUBZV at CERN. One may also use the logical name STAGE_GROUP to group users together. In this case, care must be taken to establish the correct access to the staging directories, using for example ACLs and rights identifiers. The command file to enable staging is shown below.

**Command file to enable stage access**

```
$ !
$ ! Command file to enable tape staging for a given group
$ ! Tapes are staged to disk DISK$STAGE to subdirectories
$ ! 'IN' and 'OUT' of directory 'UIC-GROUP'
$ !
$ !   Get top directory of staging disk
```

```
$ !
$  stage_disk = f$trnlnm("DISK$STAGE")
$  if f$locate(":",stage_disk) .eq. f$length(stage_disk) then -
      stage_disk = f$trnlnm(stage_disk)
$  if f$locate(".]",stage_disk) .lt. f$length(stage_disk)
$     then
$        stage_disk = stage_disk - ".]" + "]"
$     else
$        stage_disk = stage_disk + "[000000]"
$     endif
$ !
$ if f$logical("STAGE_TAPES") .eqs. "YES" then goto uic
$ write sys$output "Tape staging is currently not enabled"
$ inquire/nopunc ans "Do you wish to enable tape staging? "
$ if ans then define/system stage_tapes "YES"
$ uic:
$ type/nopage sys$input

Enter group UIC of group that you wish to enable for tape staging.

$eod
$ inquire/nopunc uic "Group UIC (RETURN=exit)? "
$ if uic .eqs. "" then exit
$ uic = uic - "[" - "]"
$ if f$locate(uic,"ALEPH,DELPHI,L3,OPAL") .lt. -
f$length("ALEPH,DELPHI,L3,OPAL") then goto lep_groups
$ number = f$identifier(uic,"NAME_TO_NUMBER")
$ if number .eq. 0 then goto bad_uic
$ group_ok:
$  in_dir  = stage_disk - "]" + ".''uic']"
$  out_dir = stage_disk - "]" + ".''uic'.OUT]"
$  write sys$output "Input directory  = ''in_dir'"
$  write sys$output "Output directory = ''out_dir'"
$ if f$search("''stage_disk'''uic'.DIR;") .nes. "" then goto exists
$ create/dir 'in_dir' /version_limit=1/owner=['uic',*] -
/protection=(s:rwe,g:rwe,w:re)
$ create/dir 'out_dir' /version_limit=1/owner=['uic',*] -
/protection=(s:rwe,g:rwe,w:re)
$ all:
$ set file/acl=(default_protection,s:rwed,o:rwed,g:rwed,w:re) -
'stage_disk''uic'.dir
$ set file/acl=(default_protection,s:rwed,o:rwed,g:rwed,w:re) -
'in_dir'out.dir
$ goto uic
$ bad_uic:
$ write sys$output ""
$ write sys$output "Tape staging is normally enabled by UIC group ", -
"and ''uic' is an invalid group UIC."
$ write sys$output "Each user in this group must ", -
"have a logical name STAGE_GROUP defined"
$ write sys$output "e.g. DEFINE STAGE_GROUP ''uic'"
$ write sys$output ""
$ inquire/nopunc ans -
"Type any character to reenter a staging group, or return to accept ''uic' "
$ if ans .eqs. "" then goto group_ok
$ goto uic
$ exists:
```

```
$ write sys$output "Group ''uic' is already enabled for staging"
$ goto uic
$ lep_groups:
$ create/dir disk$stage:['uic'] /version_limit=1/owner=id$_'uic' -
/protection=(s:rwe,g:rwe,w:re)
$ create/dir disk$stage:['uic'.OUT] /version_limit=1/owner=id$_'uic' -
/protection=(s:rwe,g:rwe,w:rwe)
$ set file/acl=(id=id$_'uic',access=read+write+execute) -
disk$stage:[000000]'uic'.dir
$ set file/acl=(id=id$_'uic',options=default,access=read+write+delete+execute) -
disk$stage:[000000]'uic'.dir
$ set file/acl=(id=id$_'uic',access=read+write+execute) -
disk$stage:['uic']out.dir
$ set file/acl=(id=id$_'uic',options=default,access=read+write+delete+execute) -
disk$stage:['uic']out.dir
$ goto uic
```

## 4.6 Multi-file staging

One may enable multi-file staging using the logical name `STAGE_FULLTAPE`. This name should point to a list of generic device types for which multi-file staging is to be enabled. When a stage request is issued for a tape of the appropriate type, all files are automatically staged.

This can be useful to prevent multiple mount requests for the same volume.

## 4.7 Multi-file concatenation

The qualifier `/CONCATENATE` may be used to stage all files on a tape to a single disk file. This option is not valid with `/RMS`.

## 4.8 Remote staging

Remote staging is currently only available through the FATMEN [3] package. However, the implementation requires the creation of special batch queues, definition of logical names and other configuration details which are best described here. The technique used to provide remote staging requires no modification to VAXTAP.

Remote staging is possible in two cases:

**1** The remote tape drive is attached to a VAX in the same VAXcluster as the client

**2** The remote tape drive is attached to a VAX that is in not in the same VAXcluster

On the client machine, a file SETUP_EXE:TPSERV.CONF must be created. This file contains a list of generic device names and the nodes on which they are served.

**An example TPSERV.CONF file**

```
TPSERV 8200 VXCRNQ

TPSERV CTRS FNALD
```

When a remote stage is required, a batch job will be submitted to a special queue on that node. If the server node is in the same VAXcluster as the client, the batch job is submitted directly. In other cases, DECnet task to task is used to start a server to submit the job and check its status.

### Configuring the batch queues

A generic batch queue is required for each generic device type that is to be served. If the queue is to be used to stage devices of type 8200, then the queue name must be STAGE_8200S. A physical queue fed by this generic queue must be created on each node that is to serve devices of this type. The job limit should be set to the number of devices that are concurrently available for this type of operation. Thus, if one has two devices of type 8200 that are dedicated to staging, the job limit should be set to 2.

### Log files from the batch jobs

The log files from the staging jobs are written to the directory defined by the system logical name STAGE_LOGS.

### Accessing the staging disk

It is assumed that the staging disk is DFS mounted on the client machine.

### Installing the stage server

On server systems which are not in the same VAXcluster as prospective clients, a small stage server must be built and defined as a DECnet object.

**DECnet definitions for the stage server**

```
MCR NCP
NCP>SET OBJECT STGSERV NUMBER 0 FILE CERN:[PRO.EXE]STGSERV.COM
NCP>DEF OBJECT STGSERV NUMBER 0 FILE CERN:[PRO.EXE]STGSERV.COM
NCP>EXIT
```

The actual command file simple runs the corresponding .EXE, and is automatically extracted as part of the FATMEN installation procedure.

**The STGSERV command file**

```
$ set noon
$!
$! STGSERV.COM
$!
$! This command file is invoked when a remote stage operation is
$! required.
$!
$! Display remote user and node name
$!
```

```
$  remote_user = f$trnlnm("SYS$REM_ID")
$  remote_node = f$trnlnm("SYS$REM_NODE")
$!
$  date_time   = f$cvtime(,"ABSOLUTE")
$!
$  write sys$output ""
$  write sys$output ">>> Start of STGSERV command file"
$  write sys$output ""
$!
$  write sys$output "Remote stage submit/query request from user ",-
"''remote_user' at ''remote_node' at ''date_time'"
$!
$! Set messages how we want them...
$!
$  SET MESSAGE/FACILITY/IDENTIFICATION/SEVERITY/TEXT
$!
$! Run the server
$!
$  RUN SETUP_EXE:STGSERV
$!
$  write sys$output ""
$  write sys$output ">>> End of STGSERV command file"
$  write sys$output ""
$!
```

The server code can be extracted with the following command file

**Extracting the stage server**

```
$ypatchy cern:[pro.src.car]fatmen.car stgserv.for :go <<!
+use,qcde.
+use,vaxvms,*stgserv.
+exe.
+pam,11,t=c.
+quit
$for stgserv.for
```

## 4.9 Server mode staging

The STAGE command is normally typed interactively, or is issued as part of a command file. An additional possibility exists, which is particularly useful in the case of multi-file tapes. This is server mode staging, where stage requests are written to a special directory and processed from there. Server mode staging requires a further logical name definition, namely STAGE_QUEUE, which should point to a directory that can be written to by users for whom server mode staging is permitted. Typically this will be handled by the use of a rights identifier.

If disk quotas are enabled on the volume to which the logical name STAGE_QUEUE points, then the directory must be set up as in the example below:

```
(IDENTIFIER=owner,ACCESS=READ+WRITE+EXECUTE+DELETE+CONTROL)
(IDENTIFIER=stage_user,ACCESS=READ+WRITE+EXECUTE)
(IDENTIFIER=owner,OPTIONS=DEFAULT,ACCESS=READ+WRITE+EXECUTE+DELETE+CONTROL)
(IDENTIFIER=stage_user,OPTIONS=DEFAULT,ACCESS=READ+WRITE+EXECUTE)
```

where owner is the owner of the directory, with a corresponding disk quota entry and stage_user the rights identifier that permits access to this directory.

If disk quotas are not enabled on the volume in question, then allowing group or world write access to the directory is sufficient.

Server mode stage requests consist of a file whose name identifies the volume and file to be staged. The file name corresponds to the normal naming convention, i.e. vsn_vid.fseq_labeltype. The contents of the file are ignored, but it can be useful to record the origin of the stage request. In the case of requests issued from FATMEN, the file contains the line

```
Stage request for vsn_vid.fseq_labeltype issued at yymmdd hhmm by user on node nodename
```

If the STAGE_QUEUE logical name is defined, then an existing staging process for the volume in question will automatically satisfy any requests in the queue. This is done as follows:

- The STAGE command checks whether the logical name STAGE_QUEUE is defined
- If so, server mode staging is automatically enabled
- It then positions the tape to each file in turn and looks to see if there are any requests for the file in question.
- If so, the request is processed and the request file deleted
- If not, the tape is positioned to the next file
- Requests are always processed by increasing file sequence number. If required, the tape will be rewound and a further pass made to satisfy new requests

The logic to decide when to launch a new staging process is embodied into the FATMEN package and is transparent to the user.

## 4.10 Cleaning the staging disk

When a stage command is issued, the stage program checks to see how much space exists on the staging disk. If there is insufficient space, it signals a cleanup process and waits until enough space has been freed.

This cleanup process should be permanently running and is started by running the STAGECLN command file, e.g. at system startup.

The STAGECLN program takes out a lock on a resource named STAGE_PURGER. It establishes a blocking AST routine which is fired whenever a user issues a STAGE command that requires more disk space than is currently available on the staging disk.

The program then spawns a subprocess which then executes the command file CLEAN_STAGE.COM. This command file performs the following actions.

**1** Illegal files are deleted from the staging disk

**2** It then deletes files older than a specified time.

**3** If enough space is recovered, it exist, otherwise the lifetime is decreased and the process repeated.

There are a number of parameters that can be set to control the action of this command file. These are shown below.

**Configuring the CLEAN_STAGE command file**

```
$ !
$ !    The following settings are for VXCERN...
$ !
$      days     = "7"      ! Start by deleting files older than DAYS days
$      hours    = "23"     ! then switch to hours when DAYS < 1
$      max_pass = 15       ! Maximum number of passes
$ !
$ !    Deletion is done on creation date. To delete according to
$ !    file usage do:
$ !
$ !    SET VOLUME/RETENTION=(1,1-) DISK$STAGE
$ !
$ !    This means that, when files are created, the expiry date is set
$ !    to now plus one day. On each access, 1 hour is added to the current
$ !    time. If this is greater than the current expiry date then the expiry
$ !    date is reset.
$ !
$ !    Then, change DELETE/CREATED/SINCE to
$ !                 DELETE/EXPIRED/BEFORE
$ !    which deletes files whose expiry date is BEFORE the specified
$ !    date & time
$ !
```

# Bibliography


[1] L. Lamport. LaTeX *A Document Preparation System.* Addison-Wesley, 1986.

[2] C. Curran et al. The FATMEN Report. Technical Report DD/89/15, CERN, 1989.

[3] J. D. Shiers. *FATMEN - Distributed File and Tape Management,* Program Library Q123. CERN, 1992.

[4] A. Cass. *CERN Tape Management System Users Guide* To be published. CERN, 1992.

[5] Frank J. Nagy. Fast Definition of DCL-like commands. Technical Report, FNAL, 1986.