Tape Subsystem Tutorial [3]

This chapter describes the following tape subsystem procedures:

- Reserving, mounting, reading, writing, and releasing a tape
- Obtaining tape status and information
- Using the tape subsystem with standard operating system commands
- Mounting ER90 volumes
- Using MLS considerations

To see an online description of a particular command or routine, use the `man`(1) command.

Note: The ER90 format is not available on systems that run the UNICOS/mk operating system or that have GigaRing support.

3.1 Getting started

To use the tape subsystem, you follow four basic steps:

1. Reserve tape resources by using the `rsv`(1) command.
2. Request a tape mount for a tape file by using the `tpmnt`(1) command.
3. Process the tape file information by using whatever commands or programs you need to accomplish what you want to do.
4. Release the tape resources reserved by using the `rls`(1) command.

The identifier of the tape resource is site configurable by the system administrator in the tape configuration file. You can use the `tpstat`(1) command to display available tape resources. The resource name is displayed in the `dgn` column.

This following tape naming conventions are used in this manual:

Name	Description
TAPE	Half-inch round tape
CART	3480 type cartridge (square tape)

QIC	Quarter-inch cartridge tape
EXB	Helical scan recording on 8mm cartridge tape
DAT	Helical scan recording on 4mm cartridge tape
SILO	3480 tape located on a SILO/400 robotic loader
WOLF	3480 tape located on a WolfCreek robotic loader
3490	3490 type
3490E	3490E type
ER90	ER90 (D-2 format) helical scan tape device

Figure 13 shows a simple example that requests a tape to be mounted. This tape has an IBM standard label with a volume serial number (VSN) of `000001` and a file name of `tf`. The contents of the disk file named `data` is copied to `tf`. After the tape file is created, the tape is unloaded and the allocated tape drive is released.

```
$ rsv CART
$ tpmnt -l sl -v 000001 -P tf -n -g CART
$ cp data tf
$ rls -a
```

Figure 13. Creating a tape

Figure 14 shows an example that will mount the previous tape, indicating that it is an old tape, and read the data from the tape into a disk file named `old.data`.

```
$ rsv CART
$ tpmnt -l sl -v 000001 -P tf -o -g CART
$ cp tf old.data
$ rls -a
```

Figure 14. Reading an existing tape file

Figure 15 shows an example that will add a new tape file (file sequence 2) to the tape 000001 and will copy the disk file named new.data into the new tape file.

```
$ rsv CART
$ tpmnt -l sl -v 000001 -q 2 -P tf -n -g CART
$ cp new.data tf
$ rls -a
```

Figure 15. Adding a new file to an existing tape

The following example shows how to submit a job through a Network Queuing System (NQS). Before submitting the job, you need to know how the NQS maps the limit specification on the qsub(1) command to the available tape device groups. To display the available device groups in limit-enforced order, issue the tprst(1) command The following example shows a listing of available device groups:

```
dev grp  w     rsvd     used available
CART             0        0    2
TAPE             0        0    0
```

In this example, there are two NQS associations. Resource group a corresponds to the CART device group and resource group b corresponds to the TAPE device group. To submit a job using the TAPE resource group, you must specify the qsub option, -lUb 1.

Figure 16 shows an example of an NQS job that requests a specific tape (SCRSL) on to which a file named data will be copied.

```
$ cat > example.sh <$ cat > example.sh <EOF
rsv TAPE
tpmnt -l sl -v SCRSL -g TAPE -P tf -n
cp data tf
rls -a
EOF
$ qsub -lUb 1 example.sh
```

Figure 16. NQS tape job

3.2 Obtaining tape status

You can use the commands and files described in this section for obtaining tape status and for sending messages to the operator.

3.2.1 Tape status commands

To check the status of your tape reservations, use the `tprst`(1) command. For example, enter the `tprst`(1) command to display the reserved-tape status device group name, number of reserved devices, number of used devices, and number of devices available for use as shown in Figure 17. In this example, no `CART`, `TAPE`, or `SILO` devices have been used or reserved, but one `TAPE` device is available for reservation:

```
$ tprst
dev grp  w     rsvd     used available
CART             0        0    0
TAPE             0        0    1
SILO             0        0    0
```

Figure 17. `tprst`(1) status display

Note: The information display in the `dev grp` column is determined by the system administrator in the tape configuration file. See Section 3.1, page 25.

To check on the status of the tape subsystem, use the `tpstat`(1) command as shown in Figure 18. To display the user ID, device group name, device name,

device identifier, device type, status of the device, job ID of the user, and volume identifier of the mounted tape, enter `tpstat`(1).

In this example, two `CART` devices, `cart120` and `cart121`, and one `TAPE` device, `tape201`, are idle and available for use. One `CART` device, `cart122`, is assigned to user `jas`, job ID `170`, with a volume identifier of `ISCSL`. It is on tape block `101`. All other devices are down and unavailable.

```
tpstat

userid  jobid dgn   a stat dvn      bx i rl ivsn   evsn   blks     NQSid
              CART  + idle cart120  04 0
              CART  + idle cart121  02 0
jas     170   CART  + assn+cart122  03 0 is ISCSL  ISCSL  101
              CART  + down cart123  05 0
              TAPE  + down tape200  10 0
              TAPE  + idle tape201  11 0
              CART  - down 300      14 0
```

Figure 18. `tpstat`(1) status display

For ER90 devices, you can specify the `-l` option on the `tpstat`(1) command to display the format identifier. This option will also display a larger block count field.

If the administrator has given you bypass label permission, you can issue a `tplist`(1) command to display the contents of a tape volume. When using `tplist`(1), you do not have to issue separate `rsv`(1), `tpmnt`(1), or `rls`(1) command. For example, to display the contents of a cartridge tape with a volume ID of `000599` and a path name of `x`, enter the `tplist`(1) command as illustrated in Figure 19.

```
tplist -v 000599 -g CART x
EBCDIC Labels
VOL1:volser:000599 owner: wek
HDR1:file_id:x           file_section:0001 file_sequence:0001
   creation date: 93148 expiration date: 93148
HDR2:max_blocksize 32768
Recsize=80 Number=3
Total Records=3, Size=240
*****TAPEMARK*****
Recsize=4096 Number=10
Total Records=10, Size=40960
*****TAPEMARK*****
EOF1: Blockcount:000010
Recsize=80 Number=2
Total Records=2, Size=160
*****TAPEMARK*****
*****TAPEMARK*****
3 file(s)
```

Figure 19. `tplist`(1) display

A message is sent to the operator to mount the cartridge. After the cartridge has been mounted, `tplist`(1) reads the cartridge and sends the output to your screen.

3.2.2 Tape log file

When you issue a `rsv`(1) command, a log file called `tape.msg` is created in your current working directory. This log file keeps track of messages the tape subsystem issues concerning your tape job. All informative and error messages are appended to this file. Figure 20 shows an example of a tape message log file.

```
Jun 14 14:00:53 0000362.1113 TM000 - tape resource reserved for you
Jun 14 14:01:04 0000373.4520 TM122 - mount tape ABCDEF(sl) ring-in, on a TAPE device
for bob 23,  () or reply cancel / device name
Jun 14 14:02:25 0000454.6664 TM048 - /tmp/jtmp.000452a/tapefile : assigned to tape203
Jun 14 14:03:26 0000515.0516 TM049 - /tmp/jtmp.000452a/tapefile : ABCDEF(sl) : open :
blocks = 0
Jun 14 14:03:28 0000516.9823 TM049 - /tmp/jtmp.000452a/tapefile : ABCDEF(sl) : bov :
write : blocks = 0
Jun 14 14:03:28 0000517.0389 TM049 - /tmp/jtmp.000452a/tapefile : ABCDEF(sl) : bof :
write : blocks = 0
Jun 14 14:03:29 0000518.3960 TM049 - /tmp/jtmp.000452a/tapefile : ABCDEF(sl) : eot :
write : blocks = 12
Jun 14 14:03:29 0000518.6526 TM049 - /tmp/jtmp.000452a/tapefile : ABCDEF(sl) : close :
        blocks = 12
Jun 14 14:03:36 0000524.7945 TM050 - tape203 : released
Jun 14 14:03:36 0000524.8156 TM029 - all tape resources released
```

Figure 20. `tape.msg`

3.2.3 Messages to operator

The `msgi`(1) and `msgr`(1) commands let you send messages to the operator. For example, the following command line sends an informative message to the operator:

1. Reserve tape resources by using the `rsv`(1) command.
2. Request a tape mount for a tape file by using the `tpmnt`(1) command.
3. Process the tape file information by using whatever commands or programs you need to accomplish what you want to do.
4. Release the tape resources reserved by using the `rls`(1) command.

```
msgi  Please check device tape00
```

To send an interactive message to the operator, use `msgr`(1). The following is an example of a message you might send to the operator. The operator may then send a reply back to you.

```
msgr  "Is tape ABC on the system?"
```

3.3 Using standard commands

This section describes some of the ways in which you may work with the tape subsystem by using standard UNICOS and UNICOS/mk commands.

Before you can issue requests to the tape subsystem for tape file processing, you must reserve the required number of tape drives for each device type needed. After you have reserved the tape drives, you may specify the tape volume in which the files to be processed are located.

After you have the volumes mounted, you can begin processing the tape files. When processing is complete, release the reserved tape drives.

3.3.1 Using the cp(1) command

The following example illustrates the process of copying a file from disk to tape using the cp(1) command:

1. Reserve a tape by using the rsv(1) command. In this example, the device group name is CART and the number of devices requested is 1:

   ```
   rsv CART 1
   ```

2. Request a tape mount by using the tpmnt(1) command. In this example, the tape has standard labels, a volume identifier of ISCSL, and a path name of /tmp/tapefile. During processing of the tpmnt(1) command, the tape subsystem creates a character-special file, /tmp/tapefile. Do not remove, rename, or move this file:

   ```
   tpmnt -v ISCSL -l sl -p /tmp/tapefile -g CART -b 32768 -n -r in
   ```

3. Copy file myfile by using the cp(1) command-line syntax, as follows:

   ```
   cp myfile /tmp/tapefile
   ```

   The cp(1) command copies bytes of data from the disk file to the tape file. It does not format any data, but blocks it into tape records of size 32768 bytes for IBM compatibles devices.

4. Release the reserved tape. The code in this example releases all resources. The tape device is allocated to you until you issue the rls(1) command with the -a, -d, or -p option or until you log out. When you issue the

`rls`(1) command or log out, the tape subsystem deletes the associated file, `/tmp/tapefile`.

```
rls -a
```

3.3.2 Using the `dd`(1) command

The following example uses the `dd`(1) command to copy a disk file to tape, converting it from ASCII to EBCDIC:

1. Reserve a tape:

   ```
   rsv TAPE 1
   ```

2. Request a tape mount by using the `tpmnt`(1) command. In this example, the tape has an IBM standard label, the volume ID is `SCRSL`, it is a new file, and a write ring is specified to be on the reel:

   ```
   tpmnt -b 4096 -l sl -v SCRSL -p /tmp/tapefile -n -r in -g TAPE
   ```

3. Use the `dd`(1) command to copy file `mydisk` to tape, specifying a block size of 4096 bytes, with a conversion from ASCII to EBCDIC for IBM compatible devices:

   ```
   dd if=mydisk of=/tmp/tapefile bs=4096 conv=ebcdic
   ```

4. You can also use the `dd`(1) command to read the tape file back into file `newfile`, and convert back to ASCII:

   ```
   dd if=/tmp/tapefile of=newfile bs=4096 conv=ascii
   ```

5. Release the tape resources:

   ```
   rls -a
   ```

3.3.3 Using the `tar`(1) command

The examples in this section show you how to read or write to tape by using the `tar`(1) command.

Procedure 1: Example 1

The following is an example of using the `tar`(1) command to read or write to tape. You must use the `-f` option of the `tar`(1) command and specify the device path name you used in the `tpmnt`(1) command.

1. Reserve a tape using the default values of the `rsv`(1) command:

   ```
   rsv TAPE 1
   ```

2. Request a tape mount by using the `tpmnt`(1) command:

   ```
   tpmnt -l sl -p /tmp/tapefile -v SCRSL -n -r in -g TAPE
   ```

3. Copy the subtree to tape, starting at the current working directory:

   ```
   tar -cvfb /tmp/tapefile 8 *
   ```

4. Change to a new directory:

   ```
   cd /tmp/newdir
   ```

5. To read the tape back in, copy the tar subtree back from tape to your current working directory:

   ```
   tar -xvfb /tmp/tapefile 8
   ```

6. Release the reserved tape:

   ```
   rls -a
   ```

Procedure 2: Example 2

The following example shows you how to read a tape that was created as shown in example 1 or on another UNIX system. In this example, the contents of the tape are read into your current working directory.

1. Reserve a tape:

   ```
   rsv TAPE 1
   ```

2. Request a tape mount by using the `tpmnt`(1) command:

   ```
   tpmnt -l sl -p /tmp/tapefile -v SCRSL -g TAPE -o
   ```

3. Read the tape, using the `tar`(1) command:

   ```
   tar -xvf /tmp/tapefile
   ```

4. Release the reserved tape:

   ```
   rls -a
   ```

3.3.4 Using the cpio(1) command

The following is an example of using the cpio(1) command to read and write to a tape. In this example, data is written to tape, then the cpio(1) command is used to read the data from tape:

1. Reserve a tape with a device group name of CART:

   ```
   rsv CART 1
   ```

2. Request a tape mount using the tpmnt(1) command:

   ```
   tpmnt -l sl -v ISCSL -p /tmp/tapefile -g CART -n -r in
   ```

3. Copy the subtree to tape, starting with the current working directory:

   ```
   find . -print | cpio -Bcov > /tmp/tapefile
   ```

4. Change to a new directory:

   ```
   cd /tmp/newdir
   ```

5. To read the tape back in, copy the cpio(1) subtree into your current working directory:

   ```
   cpio -civd < /tmp/tapefile
   ```

6. Release the reserved resources:

   ```
   rls -a
   ```

3.3.5 Using the tpmnt(1) command to read concatenated tape files

The -c option of the tpmnt(1) command allows you to read multiple tape files as though they were one tape file, with the following exceptions:

- Record size for all files to be concatenated in a job must be the same.
- Variable-length record size is not supported; it causes unpredictable results.

If front-end servicing is enabled and a tape management catalog is used, tape messages are sent to the front end.

The following example shows you how to concatenate three tape files that have the record size and block size in the label:

1. Reserve a tape by using the rsv(1) command, as follows:

   ```
   rsv TAPE 1
   ```

2. Request a tape mount by using the `tpmnt`(1) command. The first occurrence of `tpmnt`(1) establishes tape file `target`(1), to which you concatenate your tape files. If a front-end catalog does not exist, as in this example, you must specify the volume identifier by using the `-v` option. If a front-end catalog does exist, it is not necessary to specify the volume identifier. The second and third occurrences of `tpmnt`(1), using the `-f` option, specify the tape files that you want concatenated to tape file `target`(1), specified with the `-c` option.

   ```
   tpmnt -p target -l sl -o -v t03600 -g TAPE
   tpmnt -c target -l sl -o -v t03700 -f myfile.1 -g TAPE
   tpmnt -c target -l sl -o -v t03800 -f myfile.2 -g TAPE
   ```

   The tape file read contains the contents of tape files target, `myfile.1`, and `myfile.2`.

3. Copy the three tape files to disk by using the `cp`(1) command:

   ```
   cp target diskfile
   ```

4. Release the tape unit, as follows:

   ```
   rls -a
   ```

3.3.6 Using the `tpmnt`(1) command to read or write multifile tapes

Multifile volume allocation lets you process a multifile volume tape without the need for the system to unload and load tapes between files. If you use the `tpmnt`(1) command to specify that the volume identifier for the multiple-volume tape is to be first in the list of volume identifiers, the tape daemon requests that the operator mount the multifile volume tape the first time it is requested. If you request that the same volume identifier be mounted again, no mount message is sent to the operator. However, if you do not specify the same volume identifier, the request is processed for separate volumes.

When using multifile volume allocation, you can use only one file on a tape at a time; that is, you must open a specified file, process the file, and then close it before you can open another file on the same multifile tape volume. You must also reserve a device for each multifile volume tape requested (for example, if the tape files reside on several volumes, you can have files on different volumes opened at the same time; however, you must have reserved enough devices to hold all of the volumes). The following examples show you how to use multifile volume allocation.

3.3.6.1 Example 1

The following example shows you how to use multifile volume allocation to read three files from the same tape without having the operator mount the tape three times:

1. Reserve a tape:

```
rsv TAPE 1
```

2. Request a tape mount using the `tpmnt`(1) command. In this example, the tape has an ANSI standard label, the volume identifier is `SCRAL`, and the path names are `one`, `two`, and `three`, with file sequence numbers of the files to be processed as `1`, `2`, and `3`:

```
tpmnt -l al -v SCRAL -p one -q 1 -r in -g TAPE
tpmnt -l al -v SCRAL -p two -q 2 -r in -g TAPE
tpmnt -l al -v SCRAL -p three -q 3 -r in -g TAPE
```

3. Read the accessed tape files into disk files:

```
cat one > firstfile
cat two > scndfile
cat three > thrdfile
```

4. Release the reserved tape:

```
rls -a
```

3.3.6.2 Example 2

The following example shows you how to use a multifile tape to write three files to the same tape:

1. Reserve a tape:

```
rsv TAPE 1
```

2. Request a tape mount using the `tpmnt`(1) command. In this example, the tape has an ANSI standard label, the volume identifier is `SCRAL`, and the path names are `one`, `two`, and `three`, with file sequence numbers of the files to be processed as `1`, `2`, and `3`. The `-u` option is used so that the tape will not be unloaded when the process terminates. This is useful when a tape is used repeatedly, and it minimizes operator time spent mounting tapes:

```
tpmnt -u -l al -v SCRAL -p one -q 1 -n -g TAPE
tpmnt -u -l al -v SCRAL -p two -q 2 -n -g TAPE
```

```
tpmnt -u -l al -v SCRAL -p three -q 3 -n -g TAPE
```

3. Write the disk files to the specified tape files:

```
cat file1 > one
cat file2 > two
cat file3 > three
```

4. Release the reserved tape:

```
rls -a
```

3.4 Mounting ER90 volumes

This section describes how to mount volumes on an ER90 device. Use the `tpmnt`(1) command to mount a volume. If you use the `-v` option, you can specify three identifiers and a partition number to uniquely identify the requested volume.

After the system determines that the correct physical volume is mounted, it positions the tape at the requested partition. For labeled tapes, the VOL1 label is read from the tape. The identifier in the label is compared to the requested internal ID to verify that the tape was positioned to the correct logical volume.

You can bypass format ID verification by specifying the `tpmnt`(1) `-I` option. Only users with bypass label or tape manager permission may use this option.

Normal mount processing positions the tape to the beginning of a partition. To request that the volume remain at its load position, specify the `-z` option (this is useful if processing should begin near the load position). The device requires that the logical position be established before issuing any tape movement requests; therefore, you must issue a position request to an absolute track address immediately after opening the tape file when `-z` is specified. Only users with bypass label or tape manager permission may use this option. Because label processing is bypassed, the specified label type must be a bypass label.

3.5 Using MLS

Special considerations should be taken when using the tape subsystem on a UNICOS multilevel security (MLS) system. This is especially true for authorized users who are allowed access to data that their active label does not dominate.

The mandatory access control (MAC) label associated with a tape is set to the active MAC label of the process writing to the tape.

Care should be taken when writing data to tape that will later be used to restore security attributes. The MAC label on the tape should sufficiently protect the tape so that only security administrators can access the tape. The label must prevent nonadministrative users from modifying the contents of the tape.

Tape headers contain a protection flag that indicates there are additional protections on the tape. On the UNICOS operating system, this flag implies the existence of a MAC label.

On UNICOS MLS systems before UNICOS release 8.0, a MAC label is not written to the tape when the user's MAC label is level 0 with no compartments. Any other MAC label forces the protection flag to be set and the MAC label is written on the tape. Tapes without the protection flag set are assumed to be at level 0 with no compartments. UNICOS systems that do not have security enabled do not set the protection flag; hence, tapes can be moved between secure and nonsecure systems.

UNICOS 8.0 MLS requires the ability to always set the protection flag and write the MAC label on the tape. Tapes without the protection flag set cannot be accessed.

The `allow_unprotected` parameter of the `OPTIONS` statement was added for compatibility considerations. When the option is set to `YES`, the previous behavior (before UNICOS 8.0) is enforced. When it is set to `NO` (required for Trusted UNICOS), the protection flag is always set and only tapes with the protection flag set can be accessed.

Tape device groups have associated MAC label ranges. To read a tape, the MAC label on the tape must be within the MAC label range of the device group . To write to a tape, the user's active MAC label must be within the MAC label range of the device group.

If you encounter any of the following, contact your system administrator for help in resolving the issue.

- The user fails the required MAC dominate or equal restrictions when reading or writing a tape.
- The `allow_unprotected` parameter of the `OPTIONS` statement is not set to `YES`, and consequently, the user cannot access tapes created on the following:
  - A UNICOS system with security disabled
  - A system before UNICOS 8.0 by users with the active MAC label of level 0 with no compartments

  - An UNICOS system with the `allow_unprotected` parameter of the `OPTIONS` statement set to `YES` by users with the active MAC label of level 0 with no compartments

- The device group does not support the MAC label of the data being accessed through the device. If MAC restrictions are not required on the device, the MAC label ranges of the device group should be `SYSLOW` with no compartments to `SYSHIGH` with all compartments.
- If the Data Migration Facility (DMF) is installed, all device groups used by DMF must support the `SYSHIGH` MAC label.
- The `tplabel`(1) command clears the protection flag in the tape label. Systems that have the `allow_unprotected` parameter of the `OPTIONS` statement set to `NO` are not able to access the tape after the tape has been relabeled.