



Managing Tape Repairs

Three strikes, and you're out!

Peter Edwards | Sr Systems & Data Administrator

DMF Users Group | February 2017

CSIRO IMT SCIENTIFIC COMPUTING
www.csiro.au



Added after presentation to DMFUG

- After the DMFUG 2017 presentation, DMF Engineering had doubts that removing the last zone would have any effect on the next attempt to append to a tape. They checked the code and said that zone removal would not change the location of the next write, and that any observed benefits were coincidental and would probably have happened anyway. (Thanks for checking, SGI)
- The procedure and the script which implements it have therefore been changed to not include zone removal. States Strike 1 and Wait 1 still exist, and now serve the purpose of giving the tape a second chance at appendment with no other actions.
- References to zone removal have been kept in this presentation, but struck through

Tape failure modes

There are many failure modes for tape volumes

- Mount/dismount
 - includes MIR (†) issues
 - includes physical problems, like cartridge damage and leader tape issues
- Positioning & reading
- Writing from BOT
 - includes (re)writing tape labels
- Appending from EOT
- End of life flag (†)

This presentation addresses media affected by the last two.

(†) May be specific to Oracle/StorageTek T10k equipment

Append failures

Observations, based on T10k media:

- *Unless ~~spiked, spindled or folded~~ snapped, stretched or tangled by the drive, the tape media rarely fails.*
- *Recently written data is more likely to prevent subsequent appending to a tape than older data.*

This implies that if the tape is partially or fully emptied it can frequently be reused for other data with safety.

~~The simplest action is just to logically relocate the last data written, as it's most likely to be the cause of the problem. In DMF terminology, this is moving the last "zone" with *dmmove* or equivalent.~~

See first slide, after the title one.

Requiring more effort is relocating all the data. For DMF, this is "merging" or "sparsing" the tape.

Three strikes and you're out!

Several enduring states are defined:

- Normal
- Strike 1
 - Cause: DMF has had repeated append failures and has set the *hvfy* database flag
 - Action: ~~The last zone is to be relocated on a different tape~~ **Clear the flag and try again**
- Strike 2
 - Cause: The tape subsequently had *hvfy* set again due to errors
 - Action: The entire tape is to be rewritten elsewhere
- Strike 3
 - Cause: *hvfy* is set yet again - the tape is still not usable and is considered to be faulty
 - Action: It should be emptied again if necessary
- Out!
 - Cause: The tape is now empty and unavailable for further use.
 - Action: To be discarded.
- Shredder

Transitional states

More states are defined for when you're waiting for something to happen:

- Wait 1
~~The last zone has been relocated; now waiting~~
Waiting for another attempt
- Wait 2e & Wait 2p
(2e is "empty" and 2p is "partially filled")
All data has been relocated; now waiting

In both of these cases, the tape is back in full use. For now; we wait until it fills or fails again.

Filling means success; the tape is OK and returns to the Normal state.

Another failure advances to the next state in the sequence (Strike 2 & Strike 3 resp.).

Tape “end of life”

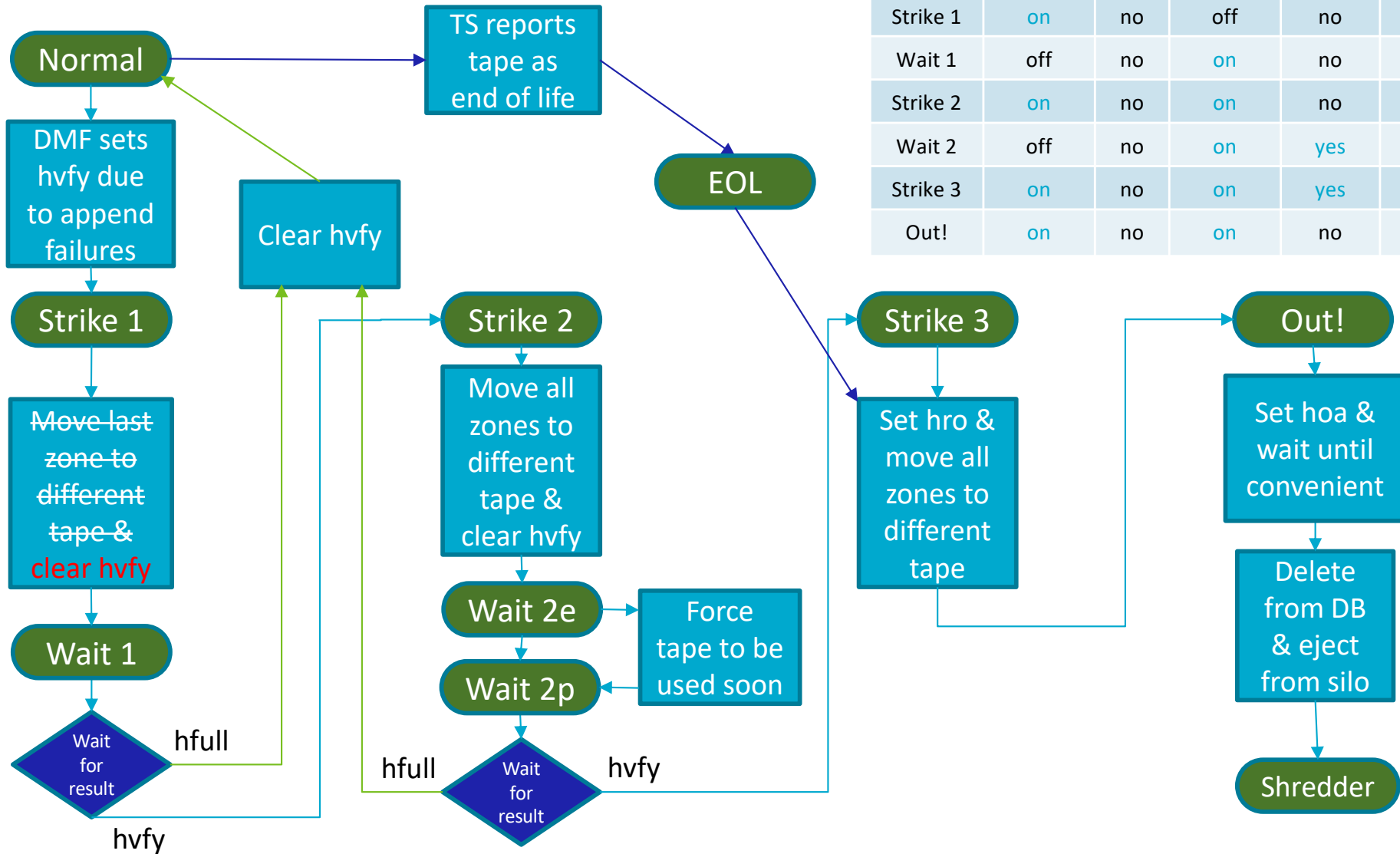
Oracle/STK considers T10k media to reach end of life after a set number of uses, even if it appears to still be reliable.

TS’s “tsreport” command reports this with two warning SCSI messages:

- The tape is nearing the end of its calculated life
- or worse,
- Media has exceeded the life pass count

Although such tapes may still be reliable for some time, it is prudent to place them into the EOL state for emptying and discarding.

State Diagram



State	hvfy DB flag	EOL file	hsite3 DB flag	Wait2 file	hoa DB flag
Normal	off	no	off	no	off
EOL	off	yes	off	off	off
Strike 1	on	no	off	no	off
Wait 1	off	no	on	no	off
Strike 2	on	no	on	no	off
Wait 2	off	no	on	yes	off
Strike 3	on	no	on	yes	off
Out!	on	no	on	no	on

Implementation (reporting)

Usage: **dm_manage_tape_repairs** [-v] {-a|-l|-x}
dm_manage_tape_repairs [-f] -s vsns...
dm_manage_tape_repairs -E
dm_manage_tape_repairs -S new-state vsns...
dm_manage_tape_repairs -R vsns...

where:

- a collates the results of previous calls to "dm_manage_tape_repairs -x" on multiple systems and writes them to stdout.
- l writes a report for the local system to stdout.
- x means report tape details for this host to files in /dmf/archive_backup instead of to stdout, for later use by "-a".
- v is verbose

Multi-host report

Report on the status of DMF tapes under repair on all systems - 2016-10-17

Totals:

Strike1	3	Multiple append failures, no repairs attempted yet
Wait1	4	EOT zone has been removed, waiting until it fills or fails again
Strike2	2	Zone removal failed to solve problem, now empty tape
Out	2	Emptied; to be deleted from DB, ejected and destroyed
In_progress	2	Actively being worked on, rather than waiting. Maybe in other states too

Details:

dmfact01	Strike1	AE001P AE001Q AL0059
dmfact01	Wait1	AE001M
dmfact01	Strike2	AE001J AE001L
dmfact01	Out	AE0032 AE003Y
dmfact01	In_progress	AE002J
dmfvic02	Wait1	VE000G
ruby	Wait1	G61503 G61816

Implementation (routine)

Usage: `dm_manage_tape_repairs [-v] {-a|-l|-x}`
`dm_manage_tape_repairs [-f] -s vsns...`
`dm_manage_tape_repairs -E`
`dm_manage_tape_repairs -S new-state vsns...`
`dm_manage_tape_repairs -R vsns...`

where:

`-s vsns`

advances tapes on the current host from their current states to the next ones

`-f` forces that advance to be done ASAP, without regard to resources required

`-E` look for end-of-life tapes in the last week's TS logs and sets their states to EOL

Examples

- Move a tape from its current state (Strike1) to the next (Wait1):

```
# dm_manage_tape_repairs -s AE001P
Changing state from Strike1 to Wait1 removing last zone
Tape AE001P now being shortened; this may take ten minutes
Done now shortened
```

- Attempting to do it again:

```
# dm_manage_tape_repairs -s AE001P
Tape AE001P is in Wait1 state, which can't be advanced manually.
Wait until either hvfy or hfull is set, which may take weeks.
```

Examples (cont'd)

- Force some data onto an empty tape

```
# dm_manage_tape_repairs -s G64027
```

Changing G64027 state from Wait2e to Wait2p by forcing data to it - takes 10 mins

```
11:47:21 Checking target tape G64027
```

```
11:47:25 Chose G60000 in se2 as data source
```

```
11:47:25 Temporarily mark all writeable se2 tapes as read-only
```

```
11:47:28 Mark all writeable T1 tapes (except G64027) read-only too
```

```
11:47:37 Flushing VG se2
```

```
11:48:37 Flushing didn't work, sparsing G60000 now
```

```
11:48:39 Waiting for G60000 to mount
```

```
11:49:18 Waiting for G64027 to mount
```

```
11:50:38 Cancel sparsing of G60000
```

```
11:50:39 Undo temporary DB changes
```

```
11:50:45 Confirm that G64027 is no longer empty
```

```
11:51:57 Done - tape G64027 is no longer empty
```

Implementation (maintenance/debugging)

Usage: `dm_manage_tape_repairs [-v] {-a|-l|-x}`
`dm_manage_tape_repairs [-f] -s vsns...`
`dm_manage_tape_repairs -E`
`dm_manage_tape_repairs -S new-state vsns...`
`dm_manage_tape_repairs -R vsns`

where:

- S new-state vsns...
sets tapes on the current host to a specific state
- R vsns...
reverts tapes on the current host back to their previous states
(it doesn't undo any data movement though)

Implementation (later additions)

Usage: `dm_manage_tape_repairs {-e|-r} vsns...`

where:

`-e vsns...`

Places specified tapes into state EOL.

`-r vsns...`

Places tapes with read errors into state Strike2, and then gently empties them, eventually ending up in state Wait2e for reuse.

These options were added after this presentation was given at DMFUG 2017.
Their logic paths are not shown in the state table in slide 7.

Does it work?

- ~~Removal of last zone and continue appending~~
 - ~~Sometimes~~
- Emptying tape and reusing
 - Frequently
- Retiring EOL tapes
 - Yes
- Automatic record keeping over a long period of time
 - Yes

References

- Local CSIRO helper scripts:
 - dm_v
 - wrapper around dmvoladm
 - dm_find_bfids
 - find BFIDs of files on a specific tape
 - dm_gently_sparse_tape
 - slowly empties a tape without DOS'ing DMF
 - dm_move_zone
 - moves a zone on one tape to another
 - dm_write_to_tape
 - forces some data onto an empty tape to make it a write candidate sooner than it would otherwise be

Thank you

CSIRO IMT Scientific Computing

Peter Edwards

Sr Systems & Data Administrator

t +61 3 9545 2377

e peter.edwards@csiro.au

w <https://wiki.csiro.au/display/ASC/Scientific+Computing+Homepage>

CSIRO IMT SCIENTIFIC COMPUTING

www.csiro.au

