Istituto Nazionale Di Fisica Nucleare
Laboratori Nazionali del Gran Sasso
S.S. 17/bis - Km 18.910 - 67010 Assergi (AQ)

Technical note for the

TAPEWATCHER

Digital VAX VMS Clusterwide Tapes Monitor

(Release 2.0)

Written by Alberto D'Ambrosio

Index

Pre-print #:

E-Mail:
VAXGS::DAMBROSIO (40618::DAMBROSIO)
dambrosio@lngs.infn.it

1. Introduction

With the VMS operating system it's impossible to get information about a specific tape unit allocated by a VAX Cluster node different from that you are connected to. More properly, it's possible to know only the name of the node the unit is allocated by, and nothing else. Morover, this information is only given by the "SHOW DEVICE/FULL" DCL command, because there aren't any system services, which can be used by an high level language,

```
Operator_ws5 » show device $1$mua15

Device                  Device           Error    Volume         Free  Trans Mnt
 Name                   Status           Count     Label        Blocks Count Cnt
$1$MUA15:      (VAXGS1)  Online alloc        0                       0      0    0
Operator_ws5 » show device/full $1$mua15

Magtape $1$MUA15: (VAXGS1), device type TA81, is online, allocated, file-
    oriented device, available to cluster, error logging is enabled.

    Error count                    0    Operations completed                 97
    Owner process                 ""    Owner UIC                         [0,0]
    Owner process ID        00000000    Dev Prot    S:RWED,O:RWED,G:RWED,W:RWED
    Reference count                0    Default buffer size                2048
    Density                     1600    Format                        Normal-11

  Device is allocated on node VAXGS1.
    Host name               "VAXGS1"    Host type, avail          VAX 8700, yes
    Alternate host name     "VAXGS2"    Alt. type, avail      VAX 6000-510, yes
    Allocation class               1

  Volume status:  no-unload on dismount, beginning-of-tape, odd parity.

Operator_ws5 »
```

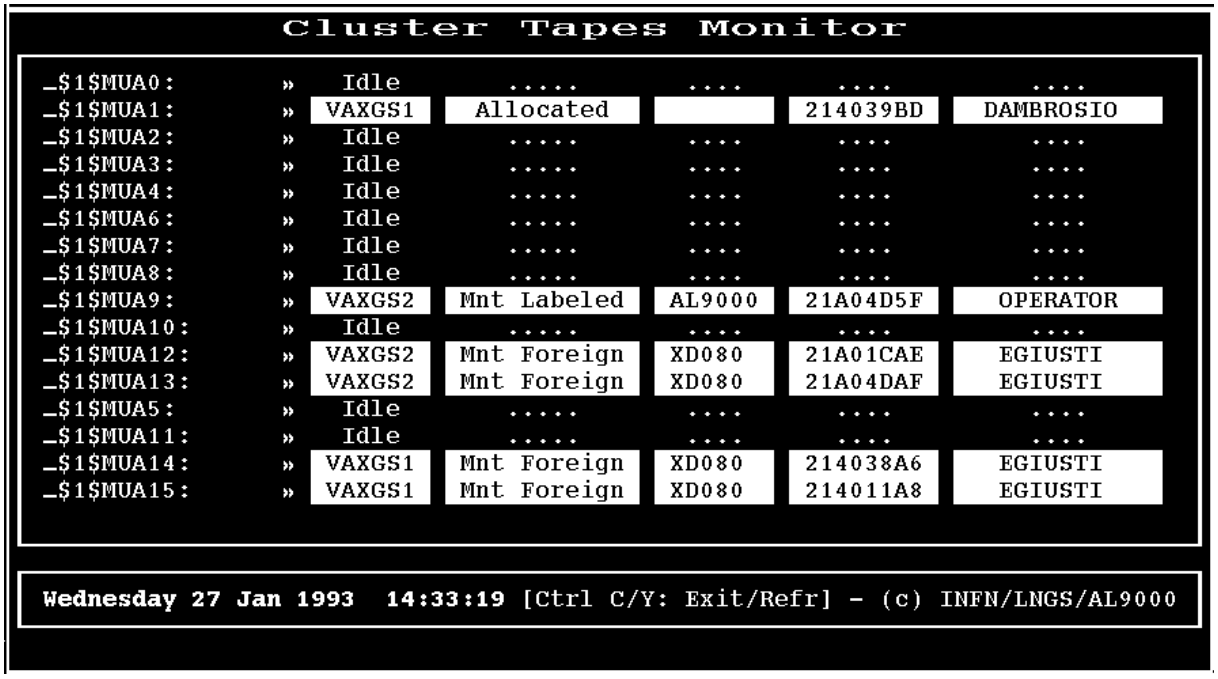
or DCL lexical function. Thus, for getting more information, it's necessary to type the "SHOW DEVICE/FULL" command, either from the remote node (by a normal interactive session) or by the SYSMAN utility.

```
Operator_vx1 » sh dev/fu $1$mua15

Magtape $1$MUA15: (HSC001), device type TA81, is online, allocated, deallocate
    on dismount, mounted foreign, record-oriented device, file-oriented device,
    served to cluster via TMSCP Server, error logging is enabled.

    Error count                    8    Operations completed             533387
    Owner process        "Gandalf_5"    Owner UIC             [CALCOLO,OPERATOR]
    Owner process ID        2140394D    Dev Prot           S:RWED,O:RWED,G:,W:
    Reference count                2    Default buffer size                 512

    Volume label            "LABEL5"    Relative volume no.                   0
    Record size                    0    Transaction count                     1
    Mount status             Process    Mount count                           1
    ACP process name              ""
    Density                     1600    Format                        Normal-11
    Host name               "HSC001"    Host type, avail              HS70, yes
    Alternate host name     "VAXGS2"    Alt. type, avail      VAX 6000-510, yes
    Allocation class               1

  Volume status:  beginning-of-tape, odd parity.

Operator_vx1 »
```

So, it's easy to understand how in a modern Computer Center, provided with many tape units, system managers and operators can be forced to type long and boring sequences of DCL commands only because they wish to know who is using a specific tape unit. The same generic user, then, could be interested in getting information about tapes used by his own batch processes, that run on cluster node queues different from the ones he is interactively connected to.

All these problems have suggested the idea of an efficient tape units monitor that gives the user the most information in the least amount of time.

TAPEWATCHER was born in this way.



Cluster Tapes Monitor

_1MUA0:	»	Idle	.....	....	....	....
_1MUA1:	»	VAXGS1	Allocated		214039BD	DAMBROSIO
_1MUA2:	»	Idle	.....	....	....	....
_1MUA3:	»	Idle	.....	....	....	....
_1MUA4:	»	Idle	.....	....	....	....
_1MUA6:	»	Idle	.....	....	....	....
_1MUA7:	»	Idle	.....	....	....	....
_1MUA8:	»	Idle	.....	....	....	....
_1MUA9:	»	VAXGS2	Mnt Labeled	AL9000	21A04D5F	OPERATOR
_1MUA10:	»	Idle	.....	....	....	....
_1MUA12:	»	VAXGS2	Mnt Foreign	XD080	21A01CAE	EGIUSTI
_1MUA13:	»	VAXGS2	Mnt Foreign	XD080	21A04DAF	EGIUSTI
_1MUA5:	»	Idle	.....	....	....	....
_1MUA11:	»	Idle	.....	....	....	....
_1MUA14:	»	VAXGS1	Mnt Foreign	XD080	214038A6	EGIUSTI
_1MUA15:	»	VAXGS1	Mnt Foreign	XD080	214011A8	EGIUSTI

Wednesday 27 Jan 1993 14:33:19 [Ctrl C/Y: Exit/Refr] - (c) INFN/LNGS/AL9000

2. Technical Characteristics.

TAPEWATCHER is a graphic display that shows the following information for each tape unit on any DEC-compatible terminal (a kind of terminal that's able to accept ANSI escape sequences):

1) The name of the unit.

2) The node the unit is used by.

3) The work status, thus if it's allocated or mounted and, in this latter case, if it's a FOREIGN or LABELED MOUNT.

4) The label of the tape (only if mounted, of course).

5) The PID of the process which allocates the unit.

6) The username associated to the PID.

The last two lines of the screen are reserved to general information:

1) Date & time.

2) Program exit and screen refreshing keys.

3) Line for possible messages related to error situations that may exceptionally happen inside the program.

The program working can be divided into three phases:

a) Initialization. During this phase the program builds the list of the units that have to be checked.

b) Visualization. It's an infinite cycle where all the units of the list are showed in groups of 16 for each video page.

c) Exit. The program automatically exits after about 20 minutes in order to save CPU time.

As the *refresh screen* command is invoked, the program restarts from the point "a" with a complete reinitialization. The scanning of the tape units list is repeated every 30 seconds. The same time occurs between a video page and the next one, independently on the fact that these ones belong to the same scanning cycle.

3. Software Analysis

In the following page, a very general flow-chart of the program is reported; it includes, over the main module, TW_CLSTWIDE.C, also a DCL task, TX_CLSTWIDE.COM. It should be noted that the software analysis is very general, mainly describing the algorithms and not their code traslation. The ***C*** has been chosen as programming language, because it's closer to the machine code than other more "human" languages (BASIC, PASCAL, etc...); morover, bearing in mind the heavy use of system services calls, it allows you to save hw resources (cpu) and execution times.

The heart of the program is the function DRIVE_SCAN (central block in the flow-chart) that, for each tape unit extracted from the list, applies the following algorithm, that considers the strange behaviour of the SYS$GETDVIW system service:

1) The unit results allocated. This means that it's used by the current node and, such a consequence, it's very easy to extract the desired information.

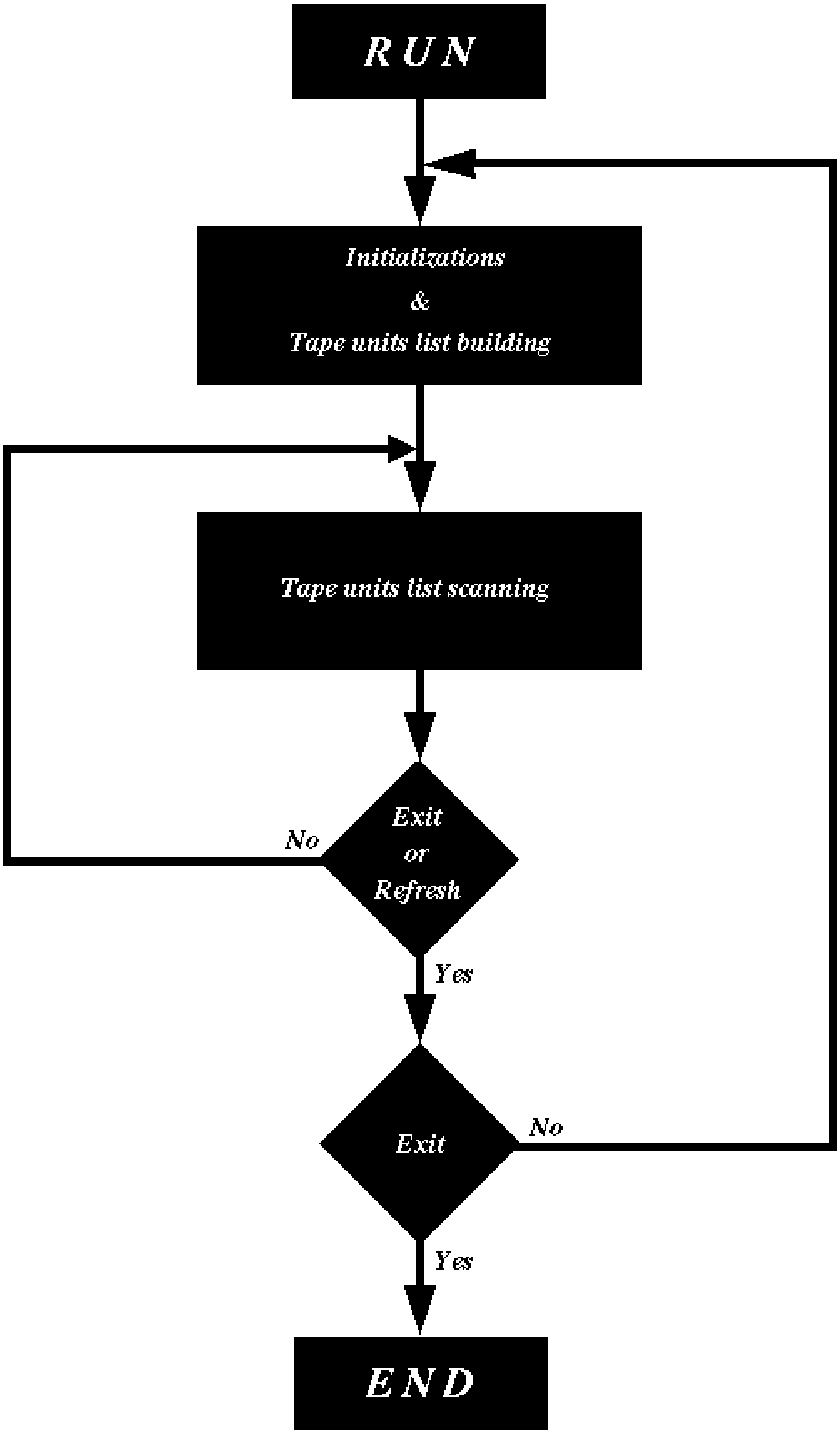
2) The unit doesn't result allocated. In this case we have two possibilities:

   2a) The unit is idle.

   2b) The unit is used by a remote node.

The SYS$DISMOU system service has been used in order to determine which of these two cases is the present one. Surprisingly, in fact, this routine has a well precise return status code in the case the unit is allocated, even if by a remote node. The use of this routine doesn't have any side effects because it's not able to DISMOUNT a unit allocated by a remote node.

The last case, the 2b, is, of course, the most complicated.; in fact, as already said in the introduction, the name of the remote node can be obtained by the extraction from the output of the "SHOW DEVICE/FULL" DCL command. The function FIND_NODE has just this assignment: it extracts the name of the remote node, that will be given to the function GET_DCHAR later. This last one performs the run of a DCL task, on the remote node located in this way; therefore, this task extracts the desired information out the remote node.


RUN
Initializations
&
Tape units list building
Tape units list scanning
Exit
or
Refresh
No
Yes
Exit
No
Yes
END

4. Installation

As first step, you must create the executable program by typing the following DCL commands:

$ CC/OPTIMIZE/NODEBUG TW_CLSTWIDE

$ DEFINE LNK$LIBRARY SYS$LIBRARY:VAXCRTL.OLB

$ LINK/NODEBUG TW_CLSTWIDE,OPTFILE/OPTIONS

$ DELETE/CONFIRM/LOG TW_CLSTWIDE.OBJ;*

where, the file OPTFILE.OPT contains only the following line:

$ SYS$SHARE:VAXCRTL/SHARE

As second step, you must copy the executable program and the related DCL task from the scratch area to a working storage area, for example:

$ COPY/CONFIRM/LOG TW_CLSTWIDE.EXE WORKDIR:*.*

$ COPY/CONFIRM/LOG TX_CLSTWIDE.COM WORKDIR:*.*

As third step, you must define a network object and the related UAF's account:

$ MC NCP

```
NCP>  DEFINE  OBJECT  TX_CLSTWIDE                              –
                      NUMBER       0                           –
                      FILE         WORKDIR:TX_CLSTWIDE.COM     –
                      PRIVILEGES   BYPASS READALL WORLD        –
                      USER         TW$SERVER                   –
                      PASSWORD     KAIZEN                      –
                      PROXY        NONE
```

This NCP command must be repeated for each node of the cluster. Morover, it will work immediately, before the next bootstrap, only if repeated with **SET** instead of **DEFINE**.

The user TW$SERVER is a normal user with, at least, the TMPMBX and NETMBX privileges. Other privileges can be set according to the ones set for the network object and the two files T%_CLSTWIDE.* placed in WORKDIR:.

As fourth and last step, you must put the following definition into the system login:

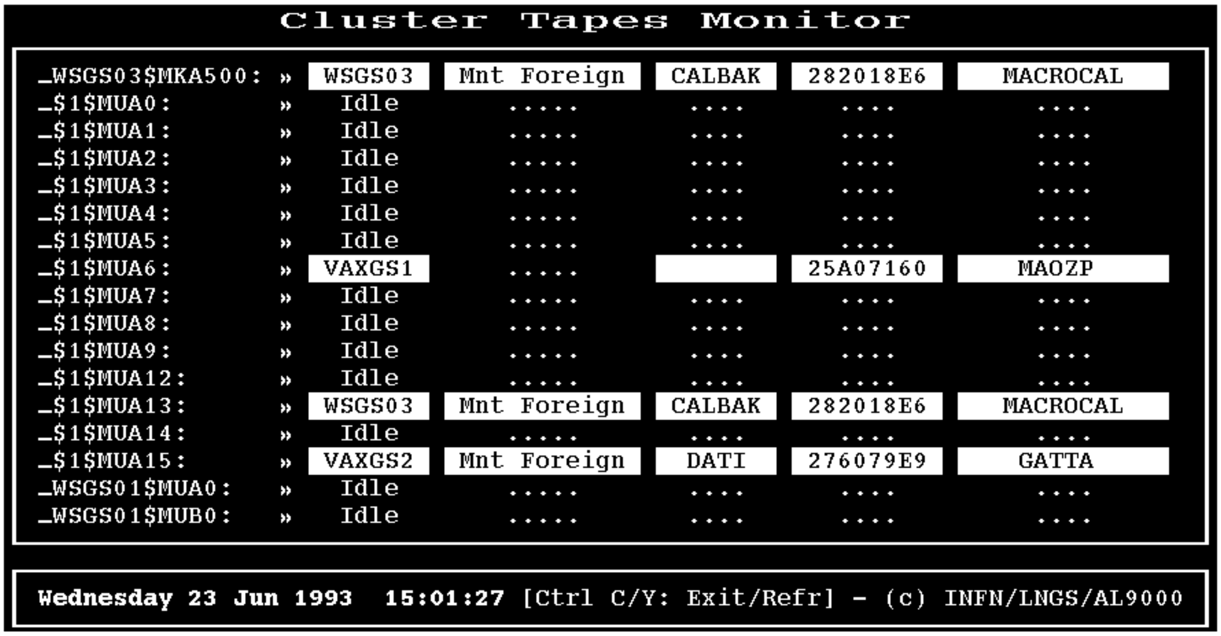
$ TAPEWATCHER :== $WORKDIR:TW_CLSTWIDE

In this way, the generic user can simply run the program by typing:

$ TAPEWATCHER

5. Known Bugs & Side Effects

1) The same user cannot run the program twice, or more, on different terminals.
2) The program creates a temporary file, SHDEV.DAT, in SYS$LOGIN:. Sometimes, this file is not deleted.
3) Users without the WORLD privilege see the "No Privilege" message in the last field of the display, for tape drives allocated by the same node they are connected to.
4) The figure of page 6 *(1. Introduction)* shows a normal visualization. As reported below, a figure shows a bad visualization where two fields are not displayed for the drive 1MUA6:. This may happen when the program that allocates the unit manages the I/O directly by QIO system routines. This is not an actual bug and, in these cases, TAPE-WATCHER displays more information than the "SHOW DEVICE" DCL command.
5) When called by a DCL menu (inside which the "ON CONTROL_Y THEN..." command is used) the program prevents the menu itself from exiting and the user loses any controls of the terminal.
6) The tape drives list is shown in non-alphabetic order.



6. Programs list