Tape Monitoring & Allocation Control System
TMACS
Reference Manual

(Version 2.1)

Legal Notices

Channel Gateway products are sold and licensed by Luminex Software, Inc. ("Luminex") pursuant to the terms and conditions of Luminex's standard Purchase and License Agreement or other applicable agreement ("Luminex Agreement"). Consult your Luminex Agreement for the specific terms and conditions governing the sale and license of your Channel Gateway.

DISCLAIMER: While every effort has been made to ensure the technical accuracy of this Manual, Luminex makes no warranties or representations, either express or implied, with respect to the contents herein. Further, should the procedures set forth in this Manual be misapplied, Luminex disclaims any warranty for damage to the product or any other liability.

Published by: Luminex Software, Inc.
Revision: April 2010
871 Marlborough Avenue
Riverside, CA 92507
Phone: 951.781.4100; Fax: 951.781.4105
info@luminex.com
www.luminex.com

LUMINEX
ENLIGHTENED | STORAGE™

Table of Contents

1 Overview

The Tape Monitoring and Allocation Control System (TMACS) intercepts tape device allocations on a z/OS system and steers them to specific groups of tape devices according to a customizable set of rules. The rules can be switched dynamically in manner similar to the MVS ACTIVATE command.

Allocations can be steered by any combination of volser range, data set name, or jobname.

TMACS runs as a functional subsystem of z/OS. No started task is required. Initial settings are loaded via a system PARMLIB TMACSxx member that is specified in the SETSSI, or keyword subsystem definition, or ZAP.

The prior version of TMACS (V1) required that the rules be defined in a "control set" which was compiled by a utility program. The control set has been replaced (in V2) by system PARMLIB members and no compile is necessary. The syntax of the rules has been changed and the scope of the rules has been expanded.

2 Allocation Rules

2.1. General Coding Syntax

- An asterisk in column 1 denotes a comment record.
- Only the first 72 characters of each line are parsed.
- Continuation lines are not supported.
- Blank lines are ignored.
- There is no method for embedding comments on the same line as a statement.
- Letter case is ignored in keywords and data parms are converted to upper case.
- Keywords can be abbreviated by removing the lower case letters in the syntax sample.
- Italicized words represent the type of information that must be placed in that location.

2.2. Device Group

The groupings of devices to which you want to steer allocations must be defined. Each rule (defined later) will steer allocations to a device group.

DEVGROUP &*devgroupname* UNIT(*unitlist*) NOTUNIT(*notunitlist*)

The *devgroupname* must have a leading ampersand (&) followed by 1-8 alpha-numeric characters.
The device numbers in *unitlist* and *notunitlist* are comma-separated and can be specific device numbers (leading / allowed if four digits). If the *unitlist* is empty then this *devgroupname* is set to 0-FFFF.
The UNIT parameter limits device selection to just the specified units. Omitting the UNIT parameter is equivalent to UNIT(/0000-/FFFF).
The NOTUNIT parameter excludes the specified units from selection. This parameter is especially useful for making sure that certain devices are not allocated.

A device group statement only pertains to the subsequent rules. Therefore, device groups that are defined at the top of the TMACS parm file will remain in effect throughout rule evaluation until that device group is redefined by a subsequent DEVGROUP statement.

If UNIT and NOTUNIT are not specified, then no tape devices are excluded for allocation for the given rule. This can be useful for opting out of TMACS allocation steering during rule evaluation.

Examples...

```
DEVGROUP &PROD1 UNIT(LUMVTS) NOTUNIT(/4908-/490F,/5908-/590F)
DEVGROUP &TEST1 UNIT(/4908-/490F,/5908-/590F)
DEVGROUP &REAL3490 UNIT(390-39F)
```

2.3. Volser Group

A volser group can be predefined and then used as one of the values in a VOL(volserlist) or NOTVOL(volserlist) selection parm of a steering rule. The volser group is especially useful when multiple rules need to refer to the same set of volsers. The volser group name must begin with an ampersand (&) followed by 1-16 characters.

VOLGROUP &*vgroupname* VOLser(*volserlist*)

selects if any of the listed volsers match. Wildcard characters in the volserlist are allowed.

In addition, volsers can be dynamically added to or removed from a volser group via operator command. Dynamic adds and removes persist, even if the rules change. However, all dynamic changes are lost whenever the system image is IPLed.

2.4. Steering Rule

If no steering rule is matched, then tape device allocation will not be steered. Each rule statement can be in either of the following accept or reject formats. Any keyword shown in mixed case means that the lower case letters are optional.

IF *selectionparms* ACCept(&*devicegroupname*)
restricts allocations to devices in the specified device group if all of the selection parms have been matched. When matched, no further rules are evaluated.

IF *selectionparms* REJect(&*devicegroupname*)
permits allocations to any device except those in the specified device group if all of the selection parms have been matched. When matched, no further rules are evaluated. This REJECT rule is an alternative to using the DEVGROUP NOTUNIT= method.

- Each "IF" statement can contain multiple selection parameters but must contain one (and only one) ACCEPT or REJECT parm.
- The IF selection parms all have to be true in order to take the ACCEPT or REJECT action.
- Only one of the values within the parentheses has to match in a positive selection parm.
- All of the values within the parentheses must NOT match in a negative (NOT...) selection parm.

OTHERWISE ACCept(&*devicegroupname*)
OTHERWISE REJect(&*devicegroupname*)
The first OTHERWISE that is encountered for a given system will define the default steering action. No further rules (for that system) will be processed.

The following selection parms can be used in any IF statement. For those parameters that allow wildcard characters, "*" means zero or more characters and "?" means one character in this position.

VOLser(*volserlist*)
selects if any of the listed volsers (or volsers with a listed volser group) match. Wildcards are allowed.

NOTVOLser(*volserlist*)
selects as long as none of the listed volsers (or volsers with a listed volser group) match. Wildcards are allowed.

JOBname(*joblist*)
selects if any of the listed job names match. Wildcards are allowed.

NOTJOBname(*joblist*)
selects as long as none of the listed job names match. Wildcards are allowed.

DSName(*dsnlist*)
selects if any of the listed data set names match. Wildcards are allowed.

NOTDSName(*dsnlist*)
selects as long as none of the listed data set names match. Wildcards are allowed.

2.5. Examples

1. Steer volsers that begin with M to Luminex VTS devices 3A0-3AF and 4A0-4AF. Be sure to allow the various generic scratch mount names on these devices and all other tape drives as well.

```
DEVGROUP &LUXVTS UNIT(3A0-3AF,4A0-4AF)
DEVGROUP &ALLDEVS UNIT(/0000-/FFFF)
VOLGROUP &SCRATCH VOL(SCRTCH,PRIVAT,MEDIA2)
IF VOL(&SCRATCH) ACCEPT(&ALLDEVS)
IF VOL(M*)   ACCEPT(&LUXVTS)
OTHERWISE    REJECT(&LUXVTS)
```

2. Steer allocations for volsers in the range P12000-P14999 to Luminex VTS devices 5C0-5DF, and volsers in the range P15000-P17999 to VTS devices 5E0-5FF. Steer allocations for volsers in the range T12000-T17999 on all other LPARs to VTS devices 600-61F. Remember to allow generic scratch mount names everywhere. Note: the LPAR statement is a planned enhancement that is not yet available. However, this example illustrates how groups and rules can be organized.

```
DEVGROUP &LUXP12 UNIT(5C0-5DF)
VOLGROUP &VOLP12 VOL(P12*,P13*,P14*)

DEVGROUP &LUXP15 UNIT(5E0-5FF)
VOLGROUP &VOLP15 VOL(P15*,P16*,P17*)

DEVGROUP &LUXT12 UNIT(600-61F)
VOLGROUP &VOLT12 VOL(T12*,T13*,T14*,T15*,T16*,T17*)

DEVGROUP &LUXALL UNIT(&LUXP12,&LUXP15,&LUXT12)
DEVGROUP &ALLDEVS UNIT(/0000-/FFFF)
VOLGROUP &SCRATCH VOL(SCRTCH,PRIVAT,MEDIA2)

*** ALL OTHER LPARS
IF VOL(&SCRATCH) ACCEPT(&ALLDEVS)
IF VOL(&VOLT12)  ACCEPT(&LUXT12)
OTHERWISE        REJECT(&LUXALL)

LPAR PROD1
IF VOL(&SCRATCH) ACCEPT(&ALLDEVS)
IF VOL(&VOLP12)  ACCEPT(&LUXP12)
IF VOL(&VOLP15)  ACCEPT(&LUXP15)
OTHERWISE        REJECT(&LUXALL)
```

3 Operator Commands

All TMACS operator commands begin with "TMACS:". Since there is no started task, the MVS modify command cannot be used. Responses to operator commands from TMACS always begin with the 4-character subsystem name. In these examples, the subsystem name is assumed to be "TMAC".

TMACS:STATus

Displays the status of TMACS.

```
TMACS:STAT
TMAC:TMACS IS ENABLED
TMAC:TMAC6002 PARMFILE=USER.PARMLIB(TMACS00)
```

TMACS:ACT=*parmsuffix*

Activates TMACS parms using the 2-character suffix that is given. The member name is "TMACS" plus the suffix. The data set containing the parm members is set by the systems programmer.

```
TMACS:ACT=00
TMAC:TMAC1000 STMT 2    DEVGROUP &EVERYDEV UNIT(0000-FFFF)
TMAC:TMAC1000 STMT 7    VOLGROUP &MIGRLOCK VOL()
TMAC:TMAC1000 STMT 8    VOLGROUP &MIGRFAIL VOL()
TMAC:TMAC1000 STMT 9    VOLGROUP &MIGRDONE VOL()
```

TMACS:ENABle
TMACS:DISAble

Enables or disables TMACS steering of tape allocations.

```
TMACS:ENAB
TMAC:TMACS ENABLED
TMAC:TMAC6002 PARMFILE=USER.PARMLIB(TMACS00)

TMACS:DISA
TMAC:TMACS DISABLED
TMAC:TMAC6002 PARMFILE=USER.PARMLIB(TMACS00)
```

TMACS:LIST,&*dynamicvolgroup*

If individual volsers are being added to or deleted from dynamic VOLGROUPs, then this command will list all dynamic volsers in a given volgroup.

TMACS:FIND,V=*volser*

Displays all dynamic VOLGROUPs that contain the given volser.

TMACS:SHOWTAP={Y|N}

Turns on a display of tape allocation parameters during each allocation in which TMACS has an opportunity to steer the allocation. These messages include the number of the TMACS statement that caused the allocation to be steered (or STMT=99999 if not steered).

TMACS:SHOWDEV={Y|N}

Turns on a display of the steps in eliminating unwanted devices from the candidates for allocation. Can be useful in solving an error in the TMACS rules.

4 Installation

1. Place the following TMACS programs into your link list and give them APF authority.
   TMACSINI TMACSDAF TMACSOPR TMACSOZZ
2. Choose a 4-character name for the subsystem. This can be TMAC or LVTS or anything else that seems appropriate. This name will appear on the front of all responses to TMACS operator commands. In the event that you have to re-intall TMACS, you'll have to pick a new name or reIPL to clear out the old name. For the remainder of these instructions, we'll assume that you picked TMAC.
3. Create a TMACS00 member in your intended TMACS parmlib. The default library is USER.PARMLIB. The library DSN can be overridden by a ZAP to TMACSINI or by specifying an initialization parm during SSI ADD. The suffix of the default TMACSxx member can be overridden by specifying an initialization parm during SSI ADD. The default TMACSxx member must exist, even if it only contains a comment line.
4. Method 1. Install the TMACS subsystem with the following operator command:
   SETSSI ADD,S=TMAC,I=TMACSINI,P=00
   Method 2. Define the TMACS subsystem in IEFSSNxx.
   SUBSYS SUBNAME(TMAC) INITRTN(TMACSINI) INITPARM(00)
   The 00 in the init parm represents the suffix of the TMACSxx member of the parmlib. It can be changed to some other value, but it may be easier to simply replace the contents of TMACS00 whenever the default settings need to be different. TMACS allows you to activate a different TMACSxx parmlib member via operator command whenever desired, but an IPL will revert back to whichever member suffix was specified during subsystem add.
5. The expanded form of the subsystem add allows you to specify the TMACS parmlib. This overrides any ZAP and the default USER.PARMLIB library. Here are expanded examples.
   ```
   SETSSI ADD,S=TMAC,I=TMACSINI,P='00,PARMLIB=TMACS.PARMLIB'
   SUBSYS SUBNAME(TMAC) INITRTN(TMACSINI)
                INITPARM('00,PARMLIB=TMACS.PARMLIB')
   ```
6. The order in which subsystems are added is significant. If multiple subsystems are handling tape allocations (SSI function code 78), then the first subsystem has the opportunity to decide before the second subsystem, and so forth. Each subsystem can selectively choose to make the final decision and prevent subsequent subsystems from participating. This is common whenever a subsystem (such as SMS or HSC) decides that the tape allocation is intended for one of the devices that that subsystem controls. Therefore, if TMACS is needed to override one of these other subsystems, then TMACS must be added prior to the add for the affected subsystem. The SMS subsystem cannot be overridden since it gets its equivalent of the 78 function directly from MVS. To see a list of subsystems that are processing SSI function 78, issue the following operator command and look for 78: **D SSI,A**
7. To define a tape device exclusively to SMS, it must be defined in the IO gen (HCD) as an ATL or MTL (library id, port id, ATL yes or MTL yes). The reverse is also true. To prevent exclusive SMS control, do not define the tape device as ATL nor MTL.