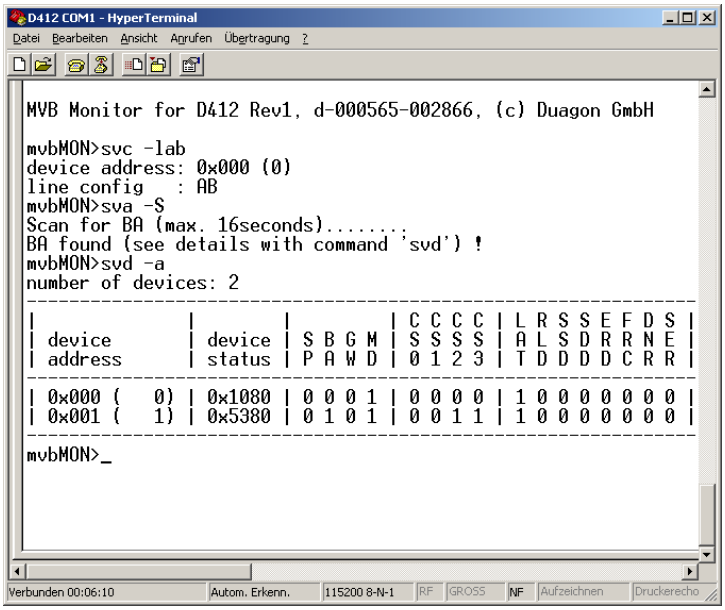
duagon

User's Guide

MVB Monitor

This document is the user's guide for the MVB monitor, a command line interface to establish MVB communication.



On having purchased products described in this user's guide, the customer acquires the right to use the products according to its specified purpose and in accordance with all operation, service and maintenance instructions. All other rights to the product, Duagon's intangible assets rights in particular, belong solely to Duagon and may not be deemed to have been assigned along with the sale of the products.

All product properties are fully described in this user's guide under express exclusion of any warranty for other properties. Of decisive relevance is this user's guide valid at the time of the order being placed. Duagon provides a warranty that the product properties are retained during the period of warranty. Evidence that the properties of the product have been retained will be brought, always and exclusively, on Duagon premises by means of a test construction pursuant to the type test.

The customer is obliged to inspect whether the products themselves are suitable for the application intended. In particular, that inspection must include the integration of the products into the intended system configuration and a check on whether the properties as per this user's guide can be fulfilled once integrated into the system configuration as planned by the customer. Since the products are not certificated for operation with security applications, the customer must take appropriate measures to ensure that any malfunctions that may occur in a system configuration with other products will be absorbed by supplementary security measures.

The period of warranty for the products is 24 months and it begins on the date the products are shipped from the factory.

The warranty that Duagon assumes for the products will, at Duagon's discretion, be limited either to the repair of or the replacement of the products at the Duagon factory. The warranty solely covers the products or parts thereof which, despite professional handling, have become defective or unusable and which arrive at the Duagon factories for repair or replacement during the period of warranty. The extent of Duagon's warranty is fully set out in this user's guide. Duagon cannot be held liable for consequential damage caused by a defect or for indirect damage or for consequential damage of any kind. Therefore the customer bears all and any costs that occur due to production downtime, for example, or due to the installation or dismantling of products or due to their transportation to Duagon and back.

Duagon's liability and warranty do not obtain if evidence cannot be brought that the products were being operated according to its specified purpose and in accordance with all operation, service and maintenance instructions as issued by Duagon.

These provisions form an integrated part of the product properties. Duagon products cannot be acquired with other or more extensive degrees of warranty and liability on the part of Duagon.

This user's guide is to be evaluated in accordance with **Swiss** law. The court of jurisdiction is the **seat of the vendor**. The applicability of the UN agreement as to international sales of goods (also known as "Viennese Purchasing Convention") is herewith expressly excluded.

For the address of duagon, please refer to the end of the document [1].

Document history

Rev.	Date	Author	Comments	Ident-Number
1	April 8th 2003	J. Rapka	1. release	d-000551-001951
2	April 13th 2004	J. Rapka	only for duagon internal use	d-000551-002458
3	January 17th 2004	J. Rapka	see appendix A "Document History"	d-000551-002683
4				
5				
6				
7				
8				
9				
10				

Table of contents

Introduction

This document is the user's guide for the MVB monitor, a command line interface to establish MVB communication.

The functionality of the MVB monitor accessible by the user is through the command line interface (CLI). The CLI is made up of a command interpreter and a command line editor. The following chapter "command set overview" categorises each of the commands and provides a brief description. The other pages describing the complete usage of each command in the MVB monitor.

Trademarks

Command Set Overview

Miscellaneous Commands

Name	Synopsis	Description
help or ?	command help	Display command set and/or command specific description.
history	command history	Display history of most recent commands.
info	application information	Display information about the application.
resett	reset monitor	Soft reset the monitor application (like first execution of monitor application).
exit	terminate monitor	Terminate monitor application.
sleep	delay execution	Delay execution for a specified number of milliseconds.

Special Commands

Name	Synopsis	Description
cm	configuration manager	Configure device (i.e. MVB control object, see command "svc") by parsing a configuration file (e.g. D2000).

Supervision Commands

Name	Synopsis	Description
svc	supervision control	Manage the MVB control object (modify/show parameters).
sva	supervision administrator	Manage MVB administrator object (BA).
svd	supervision devices	Read the MVB devices object.

Port Commands

Name	Synopsis	Description
pc	port config	Configure process data port(s) in the traffic store (add or modify).
pd	port delete	Delete process data port(s) from the traffic store.
ps	port status	Retrieve status of process data port(s).

Dataset Commands

Name	Synopsis	Description
dsp	dataset put	Put dataset to a port in the traffic store.
dsg	dataset get	Get dataset and its freshness timer from a port in the traffic store.
dsc	dataset compare	Get dataset and its freshness timer from a port in the traffic store and compare these actual values to specified nominal values.

Variable Commands

Name	Synopsis	Description
vp	variable put	Put variable to a port in the traffic store.
vg	variable get	Get variable and its freshness timer from a port in the traffic store.
vc	variable compare	Get variable and its freshness timer from a port in the traffic store and compare these actual values to specified nominal values.

TNM Commands

Name	Synopsis	Description
tnm	setup TNM connection	Setup a TNM connection, i.e. select a station by defining its network address.
tnminfo	retrieve station information	Retrieve information about a station, i.e. station inventory object, station status object, list of connected link layers (optional) and list of supported services (optional).
tnmwsr	write station reservation	This command access the reservation object, i.e. reserves or releases a station.

TNM FS Commands

Name	Synopsis	Description
tnmfspwd	print working directory name	The command prints an absolute pathname of the current working directory.
tnmfscd	change the working directory	The command causes the directory specified by the "dir" operand to become the current working directory.
tnmfsmkdir	make a directory	The command creates a new directory specified by the "dir" operand.
tnmfsrmdir	remove a directory	The command removes the directory entry specified by the "dir" operand, which must refer to an empty directory.
tnmfsls	list directory contents	The command lists information about the directory entries, i.e. name of the directory entries as well as any requested, associated information.
tnmfsrm	remove directory entry	The command removes the directory entry (i.e. file) specified by the "file" operand.
tnmfsxmodem	xmodem file transfer	The command initiates an Xmodem (Xmodem-CRC, Xmodem-1K, Ymodem) file transfer.

Miscellaneous Commands

Overview

Name	Synopsis	Description
help or ?	command help	Display command set and/or command specific description.
history	command history	Display history of most recent commands.
info	application information	Display information about the application.
resett	reset monitor	Soft reset the monitor application (like first execution of monitor application).
exit	terminate monitor	Terminate monitor application.
sleep	delay execution	Delay execution for a specified number of milliseconds.

help (?) – command help

USAGE

help [-d] [command]

DESCRIPTION

Display command set and/or command specific description.

- The string "help" or the character "?" will be seen by the monitor's command interpreter as the same command.

OPTIONS

-d

When displaying the command set, a per-command description is included.

OPERANDS

command

Command name.

EXAMPLES

Display commands set:

```
mvbMON>help
help        ?           history     info        reset       exit
sleep       cm          svc         sva         svd         pc
pd          ps          dsp         dsg         dsc         vp
vg          vc          tnm         tnminfo     tnmwsr      tnmfspwd
tnmfscd     tnmfsmkdir  tnmfsrmdir  tnmfsls     tnmfsrm     tnmfsxmodem

mvbMON>
```

Display command set with per-command description:

```
mvbMON>help -d
 help          Display command set and/or command specific…
 ?             Display command set and/or command specific…
 history       Display history of most recent commands.
…
mvbMON>
```

Display command description of command "help":

```
mvbMON>help help
Display command set and/or command specific description.
Usage:
 help  [-d] [command]
Options:
 -d       When displaying the command set, a per-command
          description is included.
Operands:
 command  Command name.
mvbMON>
```

REFERENCES

-

history – command history

USAGE

history

DESCRIPTION

Display history of most recent commands.

The monitor supports command line editing capabilities. Along with that, a history is maintained of some number of the most recently executed commands. The command line editor/history facility supports the following subset of VT100 terminal emulation keys:

Key	Action
up arrow	step back through command history
down arrow	step forward through command history
left arrow	move left one character
right arrow	move right one character
DEL	delete the character the cursor is on top of
BACKSPACE	delete the character left from the cursor

OPTIONS

-

OPERANDS

-

EXAMPLES

```
mvbMON>history
help
help -d
help help
mvbMON>
```

REFERENCES

-

info – application information

USAGE

info

DESCRIPTION

Display information about the application.

OPTIONS

-

OPERANDS

-

EXAMPLES

```
mvbMON>info
Name      : MVB Monitor
Version   : d-000565-002866
Copyright: (c) Duagon GmbH
mvbMON>
```

REFERENCES

-

reset – reset monitor

USAGE reset

DESCRIPTION Soft reset the monitor application (like first execution of monitor application).

OPTIONS -

OPERANDS -

EXAMPLES

```
mvbMON>reset
…<clear screen>

MVB Monitor, d-000565-002866, (c) Duagon GmbH

mvbMON>
```

REFERENCES -

exit – terminate monitor

USAGE exit

DESCRIPTION Terminate monitor application.

OPTIONS -

OPERANDS -

EXAMPLES

```
mvbMON>exit
…
```

REFERENCES -

sleep – delay execution

USAGE sleep time

DESCRIPTION Delay execution for a specified number of milliseconds.

OPTIONS -

OPERANDS **time**
Delay time in number of milliseconds.

EXAMPLES Delay 1 second:

```
mvbMON>sleep 1000
mvbMON>
```

REFERENCES -

Special Commands

Overview

Name	Synopsis	Description
cm	configuration manager	Configure device (i.e. MVB control object, see command "svc") by parsing a configuration file (e.g. D2000).

cm – configuration manager

USAGE

cm type filename

DESCRIPTION

Configure device (e.g. MVB control object, see command "svc") by parsing a configuration file (e.g. D2000).

NOTES:

- Before parsing the configuration file a soft reset of the MVB controller is performed, i.e.:
  - clear MVB devices object (see command "svd")
  - stop MVB administrator (see command "sva")
  - delete all process data ports from traffic store (see command "pd")
- The contents of a valid configuration file will be displayed.

OPTIONS

-

OPERANDS

type

Type of the configuration file:

- D2000 duagon D2000 configuration tool [2]
- D2000_SERVER same as above, but for SERVER configuration

filename

Name of the configuration file.

EXAMPLES

Configure device using D2000 binary file "dev_d113.bin":

```
mvbMON>cm d2000 dev_d113.bin
Parse configuration file.......
Parse configuration file.......

configuration status - device:
------------------------------
device address: 0x020 (32)
line config   : AB

configuration status - ports:
-----------------------------
number of ports: 8
0x002 (   2)  SINK    32 bytes (MVB F-code 4)
0x01E (  30)  SINK     4 bytes (MVB F-code 1)
0x123 ( 291)  SINK     2 bytes (MVB F-code 0)
0x456 (1110)  SOURCE   4 bytes (MVB F-code 1)
0x789 (1929)  SOURCE   8 bytes (MVB F-code 2)
0xABC (2748)  SOURCE  32 bytes (MVB F-code 4)
0xCBA (3258)  SOURCE  16 bytes (MVB F-code 3)
0xDEF (3567)  SINK    32 bytes (MVB F-code 4)
number of ports: 8

configuration status - bus administrator:
-----------------------------------------
BA state: D2000 (dev_d113.bin)

mvbMON>
```

REFERENCES

-

Supervision Commands

Overview

Name	Synopsis	Description
svc	supervision control	Manage the MVB control object (modify/show parameters).
sva	supervision administrator	Manage MVB administrator object (BA).
svd	supervision devices	Read the MVB devices object.

svc - supervision control

USAGE

svc [-a address][-l line]

DESCRIPTION

Manage the MVB control object (modify/show parameters).

NOTES:

- If no options/operands are specified the current values of the MVB control object are retrieved.
- Any modification of the device address (option "-a") and/or MVB input line (option "-l") will first disconnect from MVB.
  Disconnecting from MVB is like a soft reset of the MVB controller, i.e.:
  - stop MVB administrator (see command "sva")
  - clear MVB devices object (see command "svd")
  - set freshness timer of each process data port to max. value 65535ms (see commands "dsg", "dsc", "vg", "vc")
- If the current MVB input line is A, B or AB then the device will be (re-)connected to MVB.
- Finally a configured MVB administrator (BA) will be (re-)started (see command "sva").

OPTIONS

-a address

Device address: 0x000…0xFFF (0…4095).
NOTES:

- The reset value of the device address is 0x000.
- Devices with MVB bus administrator capability should use device address range 0x00…0xFF (0…255).
- Devices with MVB message data capability (e.g. TNM manager) must use device address range 0x01…0xFE (1…254).

-l line

(Lower-case ell.) MVB input line (A, B, AB or 0).
NOTES:

- The reset value of the MVB input line is 0.
- The option "-l 0" disconnect from MVB.

OPERANDS

-

EXAMPLES

Set device address to 0x001 (1):

```
mvbMON>svc -a 0x1
device address: 0x001 (1)
line config   : 0 (OFF)
mvbMON>
```

Connect to MVB using input line A:

```
mvbMON>svc -l A
device address: 0x001 (1)
line config   : A
mvbMON>
```

Set device address to 0x001 (1) and connect to MVB using input line AB:

```
mvbMON>svc -a 1 -l AB
device address: 0x001 (1)
line config   : AB
mvbMON>
```

Show MVB control object (don't modify any parameter):

```
mvbMON>svc
device address: 0x001 (1)
line config   : AB
mvbMON>
```

Disconnect from MVB (i.e. soft reset MVB controller):

```
mvbMON>svc -l 0
device address: 0x001 (1)
line config   : 0 (OFF)
mvbMON>
```

REFERENCES

See procedure "as_service_handler" using service identifier SV_MVB_SERVICE_WRITE_CONTROL in d-000487, TCN Software Architecture [3].
See also "MVB_Control" in IEC 61375-3, Multifunction Vehicle Bus [4] and "Write_MVB_Control" in IEC 61375-5, Train Network Management [5].

sva - supervision administrator

USAGE

sva [-S] [mode] [arg]

DESCRIPTION

Manage the MVB administrator object (BA).

NOTES:

- If no options/operands are specified the current status of the BA is retrieved.
- If the device is connected to MVB (see command "svc") a configured BA will be started immediately.
- The command "svc" can cause a stop and/or (re-)start of the BA.
- Any error (e.g. D2000 binary file not found) will set the BA configuration mode to OFF.

OPTIONS

-S

Scan for external BA (exclusive option).
NOTES:

- Set BA configuration mode to OFF.
- Perform a soft reset of the MVB controller by disconnecting from MVB, i.e. stop BA; clear MVB devices object (see command "svc").
- (Re-)connect to MVB and read MVB devices object repetitive (see command "svd") until its list is not empty (max. 16seconds).
- A none empty list of the MVB devices object indicates an external BA.
- Further information about BA device(s) can be retrieved by examine the MVB devices object (see command "svd").

OPERANDS

mode

BA configuration mode (reset value = OFF):

- OFF — no BA object (stop BA)
- AUTO — dynamic BA object (depends on port configuration)
- TCN — get BA object from binary file (see "MVB_Administrator" in IEC 61375-3 [4])
- D2000 — get BA object from binary file (see duagon D2000 configuration tool [2])
- D2000_SERVER — same as above, but for SERVER configuration

arg

BA configuration mode specific argument (e.g. filename for mode TCN).

EXAMPLES

Scan for external BA:

```
mvbMON>sva -S
Scan for BA (max. 16seconds).....................
No BA found !
mmvbMON>
```

Set BA configuration mode to AUTO:

```
mvbMON>sva auto
Configure dynamic BA..........
BA state: AUTO
mvbMON>
```

Set BA configuration mode to TCN using not existing file "tcn.bin":

```
mvbMON>sva tcn tcn.bin
ERROR: Can't open file.
BA state: OFF
mvbMON>
```

Set BA configuration mode to D2000 using binary file "dev_d113.bin":

```
mvbMON>sva d2000 dev_d113.bin
Parse configuration file.......
BA state: D2000 (dev_d113.bin)
mvbMON>
```

Retrieve BA state:

```
mvbMON>sva
BA state: D2000 (dev_d113.bin)
mvbMON>
```

REFERENCES

See procedure "as_service_handler" using service identifier SV_MVB_SERVICE_WRITE_ADMINISTRATOR in d-000487, TCN Software Architecture [3].
See also "MVB_Administrator" in IEC 61375-3, Multifunction Vehicle Bus [4] and "Write_MVB_Administrator" in IEC 61375-5, Train Network Management [5].

svd - supervision devices

USAGE

svd [-a|-r][-m] [address...]

DESCRIPTION

Read the MVB devices object.

The MVB devices object contains in particular the list of all devices and their device status found during devices scan ordered by increasing address, indicating for each:

- device address
- device status

OPTIONS

-a

Retrieve status of all available devices.
NOTE: No operand(s) "address" required.

-r

Retrieve status of all available devices from a device range.
NOTE: Two operands "address" are required to define the range (i.e. first address must be less than second address).

-m

Interactively provide a "more?" query to the user to support user-controlled throttling of the output.

OPERANDS

address

Device address: 0x000...0xFFF (0...4095).

EXAMPLES

List all devices found during device scan on MVB:

```
mvbMON>svd -a
number of devices: 3
-------------------------------------------------------------
|              |        |         | C C C C | L R S S E F D S |
| device       | device | S B G M | S S S S | A L S D R R N E |
| address      | status | P A W D | 0 1 2 3 | T D D D D C R R |
-------------------------------------------------------------
| 0x001 (   1) | 0x5380 | 0 1 0 1 | 0 0 1 1 | 1 0 0 0 0 0 0 0 |
| 0x010 (  16) | 0x1080 | 0 0 0 1 | 0 0 0 0 | 1 0 0 0 0 0 0 0 |
| 0x100 ( 256) | 0x0082 | 0 0 0 0 | 0 0 0 0 | 1 0 0 0 0 0 1 0 |
-------------------------------------------------------------
mvbMON>
```

List status of all available devices from device range 0x10...0x100:

```
mvbMON>svd -r 0x10 0x100
-------------------------------------------------------------
|              |        |         | C C C C | L R S S E F D S |
| device       | device | S B G M | S S S S | A L S D R R N E |
| address      | status | P A W D | 0 1 2 3 | T D D D D C R R |
-------------------------------------------------------------
| 0x010 (  16) | 0x1080 | 0 0 0 1 | 0 0 0 0 | 1 0 0 0 0 0 0 0 |
| 0x100 ( 256) | 0x0082 | 0 0 0 0 | 0 0 0 0 | 1 0 0 0 0 0 1 0 |
-------------------------------------------------------------
mvbMON>
```

List status of device 0x100 (256):

```
mvbMON>svd 0x100
-------------------------------------------------------------------
|              |        |         | C C C C | L R S S E F D S |
| device       | device | S B G M | S S S S | A L S D R R N E |
| address      | status | P A W D | 0 1 2 3 | T D D D D C R R |
-------------------------------------------------------------------
| 0x100 ( 256) | 0x0082 | 0 0 0 0 | 0 0 0 0 | 1 0 0 0 0 0 1 0 |
-------------------------------------------------------------------
mvbMON>
```

List status of device 0x20 (32):

```
mvbMON>svd 0x20
-------------------------------------------------------------------
|              |        |         | C C C C | L R S S E F D S |
| device       | device | S B G M | S S S S | A L S D R R N E |
| address      | status | P A W D | 0 1 2 3 | T D D D D C R R |
-------------------------------------------------------------------
| 0x020 (  32) |  XXXX  | X X X X | X X X X | X X X X X X X X |
-------------------------------------------------------------------
mvbMON>
```

List status of devices 0x10, 0x20 and 0x100 (device list):

```
mvbMON>svd 0x10 0x20 0x100
-------------------------------------------------------------------
|              |        |         | C C C C | L R S S E F D S |
| device       | device | S B G M | S S S S | A L S D R R N E |
| address      | status | P A W D | 0 1 2 3 | T D D D D C R R |
-------------------------------------------------------------------
| 0x010 (  16) | 0x1080 | 0 0 0 1 | 0 0 0 0 | 1 0 0 0 0 0 0 0 |
| 0x020 (  32) |  XXXX  | X X X X | X X X X | X X X X X X X X |
| 0x100 ( 256) | 0x0082 | 0 0 0 0 | 0 0 0 0 | 1 0 0 0 0 0 1 0 |
-------------------------------------------------------------------
mvbMON>
```

REFERENCES

See procedure "as_service_handler" using SV_MVB_SERVICE_READ_DEVICES as service identifier in d-000487, TCN Software Architecture [3].
See also "MVB_Devices" in IEC 61375-3, Multifunction Vehicle Bus [4] and "Read_MVB_Devices" in IEC 61375-5, Train Network Management [5].

Port Commands

Overview

Name	Synopsis	Description
pc	port config	Configure process data port(s) in the traffic store (add or modify).
pd	port delete	Delete process data port(s) from the traffic store.
ps	port status	Retrieve status of process data port(s).

pc - port config

USAGE

pc [-S][-a|-r][-t type][-s size] [address…]

DESCRIPTION

Configure process data port(s) in the traffic store (add or modify).

- A port in the traffic store is identified by its address.

OPTIONS

-S

Scan for ports and configure it as SINK (exclusive option).
NOTES:

- First delete all ports.
- Then use each port configuration (any combination of address=0x000-0xFFF, size=2/4/8/16/32 bytes) and check if a dataset was received from MVB.
- Finally configure these ports as SINK where a dataset was received from MVB.

-a

Configure all ports.
NOTE: No operand(s) "address" required.

-r

Configure a range of ports.
NOTE: Two operands "address" are required to define the range (i.e. first address must be less than second address).

-t type

Type of the port (SINK/SOURCE or 0/1).
NOTE: Assume SINK (0) if this option is omitted.

-s size

Size of the port (2/4/8/16/32 bytes or MVB F-code FC0/FC1/FC2/FC3/FC4).
NOTE: Assume 32 bytes (FC4) if this option is omitted.

OPERANDS

address

Port address: 0x000…0xFFF (0…4095).

EXAMPLES

Configure port 0x10 (16) as SINK with size 2 bytes (MVB F-code 0):

```
mvbMON>pc -s 2 0x10
mvbMON>
```

Configure port 0x18 (24) as SOURCE with size 8 bytes (MVB F-code 2):

```
mvbMON>pc -t SOURCE -s FC2 24
mvbMON>
```

Configure ports 0x110, 0x120 and 0x130 (port list) as SINK with size 4 bytes (MVB F-code 1):

```
mvbMON>pc -s 4 0x110 0x120 0x130
mvbMON>
```

Configure port range 0x200…0x220 as SOURCE with size 2 bytes (MVB F-code 0):

```
mvbMON>pc -r -t SOURCE -s 2 0x200 0x220
Configure ports...
mvbMON>
```

REFERENCES

See procedure "ap_port_manage" using command PD_PRT_CMD_CONFIG / PD_PRT_CMD_MODIFY in d-000487, TCN Software Architecture [3].

pd - port delete

USAGE

pd [-a|-r] [address...]

DESCRIPTION

Delete process data port(s) from the traffic store.

- A port in the traffic store is identified by its address.

OPTIONS

-a

Delete all ports.
NOTE: No operand(s) "address" required.

-r

Delete a range of ports.
NOTE: Two operands "address" are required to define the range (i.e. first address must be less than second address).

OPERANDS

address

Port address: 0x000...0xFFF (0...4095).

EXAMPLES

Delete port 0x10 (16):

```
mvbMON>pd 0x10
mvbMON>
```

Delete port 0x18 (24):

```
mvbMON>pd 24
mvbMON>
```

Delete ports 0x110, 0x120 and 0x130 (port list):

```
mvbMON>pd 0x110 0x120 0x130
mvbMON>
```

Delete port range 0x200...0x220:

```
mvbMON>pd -r 0x200 0x220
Delete ports...
mvbMON>
```

Delete all ports:

```
mvbMON>pd -a
Delete ports.....
mvbMON>
```

REFERENCES

See procedure "ap_port_manage" using command PD_PRT_CMD_DELETE in d-000487, TCN Software Architecture [3].

ps - port status

USAGE

ps [-a|-r][-m] [address...]

DESCRIPTION

Retrieve status of process data port(s).

- A port in the traffic store is identified by its address.

OPTIONS

-a

Retrieve status of all configured ports.
NOTE: No operand(s) "address" required.

-r

Retrieve status of all configured ports from a port range.
NOTE: Two operands "address" are required to define the range (i.e. first address must be less than second address).

-m

Interactively provide a "more?" query to the user to support user-controlled throttling of the output.

OPERANDS

address

Port address: 0x000...0xFFF (0...4095).

EXAMPLES

Retrieve status of port 0x18 (24):

```
mvbMON>ps 0x18
0x018 (  24)  SOURCE   8 bytes (MVB F-code 2)
number of ports: 1
mvbMON>
```

Retrieve status of port 0x11 (17):

```
mvbMON>ps 17
0x011 (  17)  NOT CONFIGURED
number of ports: 0
mvbMON>
```

Retrieve status of ports 0x10, 0x11and 0x18 (port list):

```
mvbMON>ps 0x10 0x11 0x18
0x010 (  16)  SINK     2 bytes (MVB F-code 0)
0x011 (  17)  NOT CONFIGURED
0x018 (  24)  SOURCE   8 bytes (MVB F-code 2)
number of ports: 2
mvbMON>
```

Retrieve status of all configured ports from port range 0x10...0x20:

```
mvbMON>ps -r 0x10 0x20
0x010 (  16)  SINK     2 bytes (MVB F-code 0)
0x018 (  24)  SOURCE   8 bytes (MVB F-code 2)
number of ports: 2
mvbMON>
```

Retrieve status of all configured ports:

```
mvbMON>ps -a
0x010 (  16)  SINK     2 bytes (MVB F-code 0)
0x018 (  24)  SOURCE   8 bytes (MVB F-code 2)
0x110 ( 272)  SINK     4 bytes (MVB F-code 1)
0x120 ( 288)  SINK     4 bytes (MVB F-code 1)
0x130 ( 304)  SINK     4 bytes (MVB F-code 1)
number of ports: 5
mvbMON>
```

REFERENCES

See procedure "ap_port_manage" using command PD_PRT_CMD_STATUS in d-000487, TCN Software Architecture [3].

Dataset Commands

Overview

Name	Synopsis	Description
dsp	dataset put	Put dataset to a port in the traffic store.
dsg	dataset get	Get dataset and its freshness timer from a port in the traffic store.
dsc	dataset compare	Get dataset and its freshness timer from a port in the traffic store and compare these actual values to specified nominal values.

dsp - dataset put

USAGE

dsp [-2|-4][-h] address [value…]

DESCRIPTION

Put dataset to a port in the traffic store.

- The traffic store can hold up to 4096 ports.
- A port in the traffic store is identified by its address.
- A port in the traffic store hold its dataset (exactly one) as it will be transmitted over the bus (big-endian number representation).
- The size of a port and its dataset can be 2/4/8/16/32 bytes.
- The dataset can be handled as an array of 8/16/32 bit values. If not otherwise specified the dataset is handled as an array of 8-bit values.
- The previous content of the dataset in a port is overwritten.

OPTIONS

-2

Handle dataset as an array of 16-bit values (operand "value").

-4

Handle dataset as an array of 32-bit values (operand "value").

-h

Assume base=16 for dataset values (operand "value"), i.e. hexadecimal without leading "0x".

OPERANDS

address

Port address: 0x000…0xFFF (0…4095).

value

Dataset value, i.e. one array element of dataset (width can be 8/16/32 bit).
NOTES:

- Max. number of accepted dataset values depends on its width and on port size.
- Dataset values which are not specified are set to 0 (e.g. size of specified dataset is less than current port size).
- If no operand "value" is specified the contents of the whole dataset is cleared, i.e. set to 0.

EXAMPLES

Put dataset "0x11, 0x22, 0x88, 0x99" (0x11 at octet offset 0, 0x22 at octet offset 1,…) to port 0x18 (24) in the traffic store (port size: 8 bytes, MVB F-code 2):

```
mvbMON>dsp 0x18 0x11 0x22 0x88 0x99
mvbMON>
…
mvbMON>dsp -h 0x18 11 22 88 99
mvbMON>
…
mvbMON>dsp 24 17 34 136 153
mvbMON>
…
mvbMON>dsp -4 24 0x11228899
mvbMON>
```

NOTE: On MVB the following frames will be transmitted for port 0x18 (24):

MF		SF
20 18		11 22 88 99 00 00 00 00

REFERENCES

See procedure "ap_put_dataset" in d-000487, TCN Software Architecture [3]; see also procedure "lp_put_dataset" in IEC 61375-2, Real-Time Protocols [6].

dsg - dataset get

USAGE

dsg [-2|-4][-d|-h] address

DESCRIPTION

Get dataset and its freshness timer from a port in the traffic store.

- The traffic store can hold up to 4096 ports.
- A port in the traffic store is identified by its address.
- A port in the traffic store hold its dataset (exactly one) as it will be transmitted over the bus (big-endian number representation).
- The size of a port and its dataset can be 2/4/8/16/32 bytes.
- The dataset can be handled as an array of 8/16/32 bit values.
- If not otherwise specified the dataset is displayed as 8-bit wide hexadecimal units (e.g. 0x00).

OPTIONS

-2

Display dataset values as 16-bit wide units.

-4

Display dataset values as 32-bit wide units.

-d

Display dataset values in decimal units.

-h

Display dataset values in hexadecimal units without leading "0x".

OPERANDS

address

Port address: 0x000…0xFFF (0…4095).

EXAMPLES

Get dataset "0x11, 0x22, 0x88, 0x99" (0x11 at octet offset 0, 0x22 at octet offset 1,…) from port 0x18 (24) in the traffic store (port size: 8 bytes, MVB F-code 2):

```
mvbMON>dsg 0x18
freshness timer:
----------------
3008ms (0x0BC0)

dataset values:
---------------
0x11 0x22 0x88 0x99 0x00 0x00 0x00 0x00
mvbMON>
…
mvbMON>dsg -d 0x18
freshness timer:
----------------
4032ms (0x0FC0)

dataset values:
---------------
 17  34 136 153   0   0   0   0
mvbMON>
…
mvbMON>dsg -2 -h 24
freshness timer:
----------------
5056ms (0x13C0)

dataset values:
---------------
1122 8899 0000 0000
mvbMON>
```

REFERENCES

See procedure "ap_get_dataset" in d-000487, TCN Software Architecture [3]; see also procedure "lp_get_dataset" in IEC 61375-2, Real-Time Protocols [6].

dsc - dataset compare

USAGE

dsc [-f freshness][-2|-4][-dh] address [value…]

DESCRIPTION

Get dataset and its freshness timer from a port in the traffic store and compare these actual values to specified nominal values.

- The traffic store can hold up to 4096 ports.
- A port in the traffic store is identified by its address.
- A port in the traffic store hold its dataset (exactly one) as it will be transmitted over the bus (big-endian number representation).
- The size of a port and its dataset can be 2/4/8/16/32 bytes.
- The dataset can be handled as an array of 8/16/32 bit values. If not otherwise specified the dataset is handled as an array of 8-bit values and displayed in hexadecimal units (e.g. 0x00).

OPTIONS

-f freshness

Nominal value of freshness timer: 0…65535 [ms].
NOTES:

- The actual value of the freshness timer must be less/equal than this nominal value.
- Assume 65535 (max. value) if this option is omitted.

-2

Handle dataset as an array of 16-bit values (operand "value").

-4

Handle dataset as an array of 32-bit values (operand "value").

-d

Display dataset values in decimal units.

-h

Assume base=16 for dataset values (operand "value"), i.e. hexadecimal without leading "0x".

OPERANDS

address

Port address: 0x000…0xFFF (0…4095).

value

Nominal dataset value, i.e. one array element of dataset (width can be 8/16/32 bit).
NOTES:

- Max. number of accepted dataset values depends on its width and on port size.
- Dataset values which are not specified are set to 0 (e.g. size of specified dataset is less than current port size).
- If no operand "value" is specified all nominal dataset values are cleared, i.e. set to 0.

EXAMPLES

Compare dataset (nominal values of dataset=0x11,0x22,0x88,0x99) with dataset of port 0x18 (port size: 8 bytes, MVB F-code 2):

```
mvbMON>dsc 0x18 0x11 0x22 0x88 0x99
freshness timer (A->actual value, N->nominal value):
-----------------------------------------------------
[A]  6016ms (0x1780)
[N] 65535ms (0xFFFF)

dataset values (A->actual value, N->nominal value):
-----------------------------------------------------
[A] 0x11 0x22 0x88 0x99 0x00 0x00 0x00 0x00
[N] 0x11 0x22 0x88 0x99 0x00 0x00 0x00 0x00
DSC OK
mvbMON>
```

Compare dataset/freshness (nominal values of dataset=12, 13; nominal value of freshness timer=10000) with dataset/freshness of port 0x18 (port size: 8 bytes, MVB F-code 2):

```
mvbMON>dsc -f 10000 -d 0x18 12 13
freshness timer (A->actual value, N->nominal value):
-----------------------------------------------------
[A]  7040ms (0x1B80)
[N] 10000ms (0x2710)

dataset values (A->actual value, N->nominal value):
-----------------------------------------------------
[A]  17  34 136 153   0   0   0   0
[N]  12  13   0   0   0   0   0   0
DSC ERROR
mvbMON>
```

REFERENCES

See commands "dsp" (dataset put) and "dsg" (dataset get).

Variable Commands

Overview

Name	Synopsis	Description
vp	variable put	Put variable to a port in the traffic store.
vg	variable get	Get variable and its freshness timer from a port in the traffic store.
vc	variable compare	Get variable and its freshness timer from a port in the traffic store and compare these actual values to specified nominal values.

vp - variable put

USAGE

vp [-b|-d] address size octet bit [value]

DESCRIPTION

Put variable to a port in the traffic store.

- The traffic store can hold up to 4096 ports.
- A port in the traffic store is identified by its address.
- A port in the traffic store hold its dataset (exactly one) as it will be transmitted over the bus (big-endian number representation).
- The size of a port and its dataset can be 2/4/8/16/32 bytes.
- A dataset can contain many variables.
- The previous value of the variable is overwritten.
- Other data of the same dataset are not affected, but consistency with them is not guaranteed.

OPTIONS

-b

Assume base=2 for operand "value", i.e. binary (e.g. 101 = 5).

-d

Assume double number representation for operand "value" (e.g. used for variable type REAL32).

OPERANDS

address

Port address: 0x000…0xFFF (0…4095).

size

Variable size (can be 1/2/4/8/16/32 bit, e.g. 1 for BOOLEAN**1**,
8 for UNSIGNED**8**, 16 for INTEGER**16**, 32 for REAL**32**).
NOTES:

- All variable sizes/types are specified in d-000487, TCN Software Architecture [3] and in IEC 61375-2, Real-Time Protocols [6].
- A variable shall be located at an offset which is a multiple of its size (alignment).

octet

Octet offset (first octet has offset 0, max. octet offset depends on port size and variable size).

bit

Bit number (counted from the right, max. bit number depends on variable size/type, e.g. 0…7 for BOOLEAN1, 0/2/4/6 for ANTIVALENT2, 0/4 for BCD4/ENUM4, all other 0 due to alignment).

value

Variable value (width depends on variable size, can be 1/2/4/8/16/32 bit).
NOTE: If operand "value" is omitted the variable is cleared, i.e. set to 0.

EXAMPLES

Put variable of type BOOLEAN1 (variable_size=1, octet_offset=5, bit_number=3, variable_value=1) to port 0x18 (24) in the traffic store (port size: 8 bytes, MVB F-code 2):

```
mvbMON>vp 0x18 1 5 3 1
mvbMON>
```

NOTE: On MVB the following frames will be transmitted for port 0x18 (24):

MF		SF
20 18		11 22 88 99 00 08 00 00

Put variable of type UNSIGNED8 (variable_size=8, octet_offset=7, bit_number=0, variable_value=101b=5) to port 0x18 (24) in the traffic store (port size: 8 bytes, MVB F-code 2):

```
mvbMON>vp -b 0x18 8 7 0 101
mvbMON>
```

NOTE: On MVB the following frames will be transmitted for port 0x18 (24):

MF		SF
20 18		11 22 88 99 00 08 00 05

REFERENCES

See procedure "ap_put_variable" in d-000487, TCN Software Architecture [3] and in IEC 61375-2, Real-Time Protocols [6].

vg - variable get

USAGE

vg address size octet bit

DESCRIPTION

Get variable and its freshness timer from a port in the traffic store.

- The traffic store can hold up to 4096 ports.
- A port in the traffic store is identified by its address.
- A port in the traffic store hold its dataset (exactly one) as it will be transmitted over the bus (big-endian number representation).
- The size of a port and its dataset can be 2/4/8/16/32 bytes.
- A dataset can contain many variables.

OPTIONS

-

OPERANDS

address

Port address: 0x000…0xFFF (0…4095).

size

Variable size (can be 1/2/4/8/16/32 bit, e.g. 1 for BOOLEAN**1**, 8 for UNSIGNED**8**, 16 for INTEGER**16**, 32 for REAL**32**).
NOTES:

- All variable sizes/types are specified in d-000487, TCN Software Architecture [3] and in IEC 61375-2, Real-Time Protocols [6].
- A variable shall be located at an offset which is a multiple of its size (alignment).

octet

Octet offset (first octet has offset 0, max. octet offset depends on port size and variable size).

bit

Bit number (counted from the right, max. bit number depends on variable size/type, e.g. 0…7 for BOOLEAN1, 0/2/4/6 for ANTIVALENT2, 0/4 for BCD4/ENUM4, all other 0 due to alignment).

EXAMPLES

Get variable of type UNSIGNED16 (variable_size=16, octet_offset=2, bit_number=0) from port 0x18 (24) in the traffic store (port size: 8 bytes, MVB F-code 2):

```
mvbMON>vg 0x18 16 2 0
freshness timer:
----------------
7040ms (0x1B80)

variable value:
---------------
binary  : 1000 1000 1001 1001
hex     : 0x8899
unsigned: 34969
signed  : -30567
mvbMON>
```

REFERENCES

See procedure "ap_get_variable" in d-000487, TCN Software Architecture [3] and in IEC 61375-2, Real-Time Protocols [6].

vc - variable compare

USAGE

vc [-f freshness][-bd] address size octet bit [value]

DESCRIPTION

Get variable and its freshness timer from a port in the traffic store and compare these actual values to specified nominal values.

- The traffic store can hold up to 4096 ports.
- A port in the traffic store is identified by its address.
- A port in the traffic store hold its dataset (exactly one) as it will be transmitted over the bus (big-endian number representation).
- The size of a port and its dataset can be 2/4/8/16/32 bytes.
- A dataset can contain many variables.

OPTIONS

-f freshness

Nominal value of freshness timer: 0…65535 [ms].
NOTES:

- The actual value of the freshness timer must be less/equal than this nominal value.
- Assume 65535 (max. value) if this option is omitted.

-b

Assume base=2 for operand "value", i.e. binary (e.g. 101 = 5).

-d

Assume double number representation for operand "value" (e.g. used for variable type REAL32).

OPERANDS

address

Port address: 0x000…0xFFF (0…4095).

size

Variable size (can be 1/2/4/8/16/32 bit, e.g. 1 for BOOLEAN**1**,
8 for UNSIGNED**8**, 16 for INTEGER**16**, 32 for REAL**32**).
NOTES:

- All variable sizes/types are specified in d-000487, TCN Software Architecture [3] and in IEC 61375-2, Real-Time Protocols [6].
- A variable shall be located at an offset which is a multiple of its size (alignment).

octet

Octet offset (first octet has offset 0, max. octet offset depends on port size and variable size).

bit

Bit number (counted from the right, max. bit number depends on variable size/type, e.g. 0…7 for BOOLEAN1, 0/2/4/6 for ANTIVALENT2, 0/4 for BCD4/ENUM4, all other 0 due to alignment).

value

Nominal value of variable (width depends on variable size, can be 1/2/4/8/16/32 bit).
NOTE: If operand "value" is omitted the nominal value of the variable is cleared, i.e. set to 0.

EXAMPLES

Compare variable (nominal value of variable=101b=5) with variable from port 0x18 (variable is of type UNSIGNED8; variable_size=8, octet_offset=7, bit_number=0):

```
mvbMON>vc -b 0x18 8 7 0 101
freshness timer (A->actual value, N->nominal value):
-----------------------------------------------------
[A]  8000ms (0x1F40)
[N] 65535ms (0xFFFF)

variable value (A->actual value, N->nominal value):
-----------------------------------------------------
[A] binary  : 0000 0101
[N] binary  : 0000 0101
[A] hex     : 0x05
[N] hex     : 0x05
[A] unsigned: 5
[N] unsigned: 5
[A] signed  : 5
[N] signed  : 5
VC OK
mvbMON>
```

Compare variable/freshness (nominal value of variable=0x1234; nominal value of freshness timer=5000) with variable/freshness from port 0x18 (variable is of type UNSIGNED8; variable_size=8, octet_offset=7, bit_number=0):

```
mvbMON>vc -f 5000 0x18 16 2 0 0x1234
freshness timer (A->actual value, N->nominal value):
-----------------------------------------------------
[A]  9024ms (0x2340)
[N]  5000ms (0x1388)

variable value (A->actual value, N->nominal value):
-----------------------------------------------------
[A] binary  : 1000 1000 1001 1001
[N] binary  : 0001 0010 0011 0100
[A] hex     : 0x8899
[N] hex     : 0x1234
[A] unsigned: 34969
[N] unsigned: 4660
[A] signed  : -30567
[N] signed  : 4660
VC ERROR
mvbMON>
```

REFERENCES

See commands "vp" (variable put) and "vg" (variable get).

TNM Commands

Overview

Name	Synopsis	Description
tnm	setup TNM connection	Setup a TNM connection, i.e. select a station by defining its network address.
tnminfo	retrieve station information	Retrieve information about a station, i.e. station inventory object, station status object, list of connected link layers (optional) and list of supported services (optional).
tnmwsr	write station reservation	This command access the reservation object, i.e. reserves or releases a station.

The MVB monitor can handle only one TNM connection at time. Before executing any TNM commands, a TNM connection must be setup by command "tnm", which select a station for TNM access.

tnm – setup TNM connection

USAGE

tnm [-S][-n node][-s station][-t topo]

DESCRIPTION

Setup a TNM connection, i.e. select a station by defining its network address.

NOTES:

- If no options are specified the current selected network address is retrieved.
- The TNM service "Read_Station_Station" is used to check, if a station has TNM capability, i.e. a TNM agent is running and processing incoming TNM services.

OPTIONS

-S

Scan MVB bus for stations with TNM capability (exclusive option).
NOTES:

- All MVB devices in the device address range 0x01…0xFE (1…254), which have message data capability (see MVB device object; command "svd"), will be checked for TNM capability (assuming Station_Id=Device_Address).

-n node

Node_Address: 0x00…0x3F (0…63).
NOTES:

- The reset value of the Node_Address is 0x00.
- Node_Address 0x00=SAME_NODE (not going over train bus).
- If Node_Address<>SAME_NODE, then Topo_Counter<>ANY_TOPO is required.

-s station

Station_Id: 0x00…0xFF (0…255).
NOTES:

- The reset value of the Station_Id is 0x00.
- On most MVB stations, the Station_Id is equal to the MVB Device_Address (only for range 0x01…0xFE).
- Station_Id 0x00=SAME_STATION (local Agent).

-t topo

Topo_Counter: 0x00…0x3F (0…63).
NOTES:

- The reset value of the Topo_Counter is 0x00.
- Topo_Counter 0x00=ANY_TOPO (not going over train bus).
- If Node_Address<>SAME_NODE, then Topo_Counter<>ANY_TOPO is required.

OPERANDS

-

EXAMPLES

Scan MVB bus for stations with TNM capability:

```
mvbMON>tnm -S
-------------------------------------------------------
| StationID  | BusID | DeviceAddress  | StationStatus |
-------------------------------------------------------
| 0x07 (  7) |     0 | 0x0007 (    1) | 0x1040        |
| 0x30 ( 48) |     0 | 0x0030 (   48) | 0x5000        |
-------------------------------------------------------
mmvbMON>
```

Show address of current selected station (NOTE: example shows reset values):

```
mvbMON>tnm
node   : 0x00 (  0) SAME_NODE
station: 0x00 (  0) SAME_STATION
topo   : 0x00 (  0) ANY_TOPO
mvbMON>
```

Select station (e.g. station=7 on same MVB bus):

```
mvbMON>tnm -s 7
node   : 0x00 (  0) SAME_NODE
station: 0x07 (  7)
topo   : 0x00 (  0) ANY_TOPO
mvbMON>
```

REFERENCES

See "AM_ADDRESS" in d-000487, TCN Software Architecture [3] and in IEC 61375-2, Real-Time Protocols [6].
See also procedure "as_service_handler" using SV_MVB_SERVICE_READ_DEVICES as service identifier in d-000487, TCN Software Architecture [3].
See also "MVB_Devices" in IEC 61375-3, Multifunction Vehicle Bus [4].
See also "Read_Station_Status" in IEC 61375-5, Train Network Management [5].

tnminfo – retrieve station information

USAGE

tnminfo [-sl][-m]

DESCRIPTION

Retrieve information about a station, i.e. station inventory object, station status object, list of connected link layers (optional) and list of supported services (optional).

NOTES:

- The TNM service "Read_Station_Inventory" is used to read the station inventory object.
- The TNM service "Read_Station_Status" is used to read the station status object.

OPTIONS

-s

List supported services and their description text.
NOTES:

- The value "service_set" returned by TNM service "Read_Station_Inventory" is handled as an array of bytes. The first bit (MSB=most significant bit) of the first byte (MSB=most significant byte) corresponds to SIF_code0.
- The TNM service "Read_Service_Descriptor" is used to read the description text of a service.
- If the description text read by TNM service "Read_Service_Descriptor" is empty, the default description text is used (e.g. SIF_code0=Read_Station_Status, SIF_code10=Read_MVB_Status, SIF_code128=User_Service_0).

-l

List connected link layers and their description text (i.e. type and name).
NOTES:

- The value "link_set" returned by TNM service "Read_Station_Inventory" contains one bit for each connected link layer; the first bit (MSB=most significant bit) corresponds to link_layer0 (e.g. 8000h=link_layer0, 0001h=link_layer15).
- The TNM service "Read_Links_Descriptor" is used to read the description text of a connected link layer.
- The type of a link layer is a 8-bit number. The following types are known:

Type	Description Text
00h	unknown link
01h	MVB
02h	WTB
03h	memory mailbox
04h	serial link
05h	CAN bus
06h-FFh	unknown link

-m

Interactively provide a "more?" query to the user to support user-controlled throttling of the output.

OPERANDS

-

EXAMPLES

Show station inventory object:

```
mvbMON>tnminfo
agent version       : IEC 61375-5; TCN, Clause 5: TNM
manufacturer version: duagon GmbH, CH-8953 Dietikon
device type         : Babylon MVB Interface
service set         : FA3C000200F00000000000000000000000
                      FFFF8000000000000000000000000000
link set            : 8001
station ID          : 0x07 (7)
station status      : 0x1040
station name        :
bus ID              : 0
device address      : 0x0007 (7)
mmvbMON>
```

Show station inventory object with list of connected link layers and list of supported services:

```
mvbMON>tnminfo -l -s
agent version       : IEC 61375-5; TCN, Clause 5: TNM
manufacturer version: duagon GmbH, CH-8953 Dietikon
device type         : Babylon MVB Interface
service set         : FA3C000200F00000000000000000000000
                      FFFF8000000000000000000000000000
link set            : 8001
station ID          : 0x07 (7)
station status      : 0x1040
station name        :
bus ID              : 0
device address      : 0x0007 (7)

List of supported services:
-------------------------------------------------------------
| SIF | Service_Description                                   |
-------------------------------------------------------------
|   0 | Read_Station_Status                                   |
|   1 | Write_Station_Control                                 |
|   2 | Read_Station_Inventory                                |
|   3 | Write_Reservation                                     |
|   4 | Read_Service_Descriptor                               |
…snip…
| 128 | FS_access                                             |
| 129 | FS_open                                               |
…snip…
| 199 | User_Service_71                                       |
…snip…
| 203 | This is an extra long description text, which did not |
|     | fit into one line.                                    |
…snip…
-------------------------------------------------------------

List of connected link layers:
-------------------------------------------------------------
| ID | TYPE               | NAME                            |
-------------------------------------------------------------
|  0 | 01 MVB             | MVB d113L-MDFULL                |
| 15 | 00 unknown link    |                                 |
-------------------------------------------------------------
mmvbMON>
```

REFERENCES

See "Read_Station_Inventory, Read_Station_Status, Read_Service_Descriptor, Read_Links_Descriptor" in IEC 61375-5, Train Network Management [5].

tnmwsr – write station reservation

USAGE

tnmwrs [-t time_out][-m manager_id][-o] command

DESCRIPTION

This command access the reservation object, i.e. reserves or releases a station.

OPTIONS

-t time_out

Time during which the station remains reserved: 1…3600 seconds.
NOTES:

- Must be specified if command=RESERVE.

-m manager_id

Identifies the manager: 0x00000000…0xFFFFFFFF (0…4294967295).
NOTES:

- Only required if command=RESERVE.
- Assume 0 if this option is omitted.

-o

Identifies the access type. If the option is set, then override access requested (OVERRIDE=1), otherwise write access requested (WRITEREQ=0).

OPERANDS

command

One of the following values:

Symbolic Constant	Decimal Number	Remarks
RESERVE	1	reserve the station for this manager
KEEPREL	2	release and keep changes
STARTREL	3	release and restart

EXAMPLES

Reserve station with time out of 60 seconds:

```
mvbMON>tnmwsr -t 60 RESERVE
mvbMON>
```

Reset station, regardless of the current reservation status:

```
mvbMON>tnmwsr -o 3
mvbMON>
```

REFERENCES

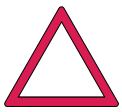
See "Write_Station_Reservation" in IEC 61375-5, Train Network Management [5].

TNM FS Commands

Overview

The TNM file system commands offers the capability to manage the file system of a station by using TNM services.

Name	Synopsis	Description
tnmfspwd	print working directory name	The command prints an absolute pathname of the current working directory.
tnmfscd	change the working directory	The command causes the directory specified by the "dir" operand to become the current working directory.
tnmfsmkdir	make a directory	The command creates a new directory specified by the "dir" operand.
tnmfsrmdir	remove a directory	The command removes the directory entry specified by the "dir" operand, which must refer to an empty directory.
tnmfsls	list directory contents	The command lists information about the directory entries, i.e. name of the directory entries as well as any requested, associated information.
tnmfsrm	remove directory entry	The command removes the directory entry (i.e. file) specified by the "file" operand.
tnmfsxmodem	xmodem file transfer	The command initiates an Xmodem (Xmodem-CRC, Xmodem-1K, Ymodem) file transfer.



It depends on the TNM managed file system, if a pathname must be given in POSIX-style (e.g. an absolute pathname begins with a slash "/") or DOS/Windows-style (e.g. an absolute pathname begins with a drive letter such as "C:\").

Most of the functionality of these TNM file system commands depends on the TNM managed file system (e.g. directories, hard/symbolic links, permissions, timestamp, etc).

tnmfspwd – print working directory name

USAGE

tnmfspwd

DESCRIPTION

The command prints an absolute pathname of the current working directory.

- It depends on the TNM managed file system, if the absolute pathname is displayed in POSIX-style (i.e. begin with a slash "/") or DOS/Windows-style (i.e. begin with a drive letter such as "C:\").

OPTIONS

-

OPERANDS

-

EXAMPLES

Prints the absolute pathname (POSIX-style) of the current working directory:

```
mvbMON>tnmfspwd
/bin
mvbMON>
```

Prints the absolute pathname (DOS/Windows-style) of the current working directory:

```
mvbMON>tnmfspwd
C:\bin
mvbMON>
```

REFERENCES

See TNM service "TNM_FS_getcwd" in d-000487, TCN Software Architecture [3] and in IEC 61375-2, Real-Time Protocols [6].
See also UNIX utility "pwd" and POSIX procedure "getcwd" in "The Single UNIX Specification, Version 3" [8].

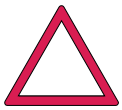
tnmfscd – change the working directory

USAGE

tnmfscd dir

DESCRIPTION

The command causes the directory specified by the "dir" operand to become the current working directory, that is, the starting point for path searches for relative pathnames (e.g. a relative POSIX-style pathname did not begin with a slash "/").



Use this command with care, since the managed station may have only one process (e.g. d113 using eCos) and therefore the same working directory is used by the TNM agent and the user application.

OPTIONS

-

OPERANDS

dir

An absolute or relative pathname of the directory that becomes the new working directory.

EXAMPLES

Change the working directory to "/usr" specifying the absolute pathname "/usr":

```
mvbMON>tnmfscd /usr
/usr
mvbMON>
```

Display the absolute pathname (POSIX-style) of the current working directory and change the working directory to "/usr" specifying the relative pathname "usr":

```
mvbMON>tnmfspwd
/
mvbMON>tnmfscd usr
/usr
mvbMON>
```

REFERENCES

See TNM service "TNM_FS_chdir" in d-000487, TCN Software Architecture [3] and in IEC 61375-2, Real-Time Protocols [6].
See also UNIX utility "cd" and POSIX procedure "chdir" in "The Single UNIX Specification,. Version 3" [8].

tnmfsmkdir – make a directory

USAGE

tnmfsmkdir [-m mode] dir

DESCRIPTION

The command creates a new directory specified by the "dir" operand.

OPTIONS

-m mode

Set the file permission bits of the newly-created directory to the specified "mode" value.
The file permission bits are specified by a non-negative octal number. For each bit set in the octal number, the corresponding file permission bit shown in the following table will be set; all other file permission bits will be cleared.

Octal	Mode bit	Octal	Mode bit	Octal	Mode bit	Octal	Mode bit
4000	S_ISUID	0400	S_IRUSR	0040	S_IRGRP	0004	S_IROTH
2000	S_ISGID	0200	S_IWUSR	0020	S_IWGRP	0002	S_IWOTH
1000	S_ISVTX	0100	S_IXUSR	0010	S_IXGRP	0001	S_IXOTH
		0700	S_IRWXU	0070	S_IRWXG	0007	S_IRWXO

NOTE: Assume file mode 0777 (i.e. the value of the bitwise inclusive OR of S_IRWXU, S_IRWXG and S_IRWXO) if this option is omitted.

OPERANDS

dir

An absolute or relative pathname of a directory to be created.

EXAMPLES

Change the working directory to "/" specifying the absolute pathname "/". Create the directory "/mvb" specifying the absolute pathname "/mvb":

```
mvbMON>tnmfscd /
/
mvbMON>tnmfsmkdir /mvb
mvbMON>
```

Change the working directory to "/" specifying the absolute pathname "/". Create the directory "/privat" with file permission bits S_IRWXU specifying the relative pathname "privat":

```
mvbMON>tnmfscd /
/
mvbMON>tnmfsmkdir -m 700 privat
mvbMON>
```

REFERENCES

See TNM service "TNM_FS_mkdir" in d-000487, TCN Software Architecture [3] and in IEC 61375-2, Real-Time Protocols [6].
See also UNIX utility "mkdir" and POSIX procedure "mkdir" in "The Single UNIX Specification,. Version 3" [8].

tnmfsrmdir – remove a directory

USAGE

tnmfsrmdir dir

DESCRIPTION

The command removes the directory entry specified by the "dir" operand, which must refer to an empty directory.

- The user must confirm the execution of the command ("continue?" query).

OPTIONS

-

OPERANDS

dir

An absolute or relative pathname of an empty directory to be removed.

EXAMPLES

Remove the directory "/mvb" specifying the absolute pathname "/mvb":

```
mvbMON>tnmfsrmdir /mvb
continue? [y/n]
mvbMON>
```

REFERENCES

See TNM service "TNM_FS_rmdir" in d-000487, TCN Software Architecture [3] and in IEC 61375-2, Real-Time Protocols [6].
See also UNIX utility "rmdir" and POSIX procedure "rmdir" in "The Single UNIX Specification,. Version 3" [8].

tnmfsls – list directory contents

USAGE

tnmfsls [-a][-l|-s][-m] [file]

DESCRIPTION

The command lists information about the directory entries, i.e. name of the directory entries as well as any requested, associated information.

OPTIONS

-a

Write out all directory entries, including those whose names begin with a period (".").

-l

(Lower-case ell.) Use a POSIX-style long listing format.

-s

Use a file system specific long listing format.

-m

Interactively provide a "more?" query to the user to support user-controlled throttling of the output.

OPERANDS

file

An absolute or relative pathname of a directory entry (e.g. file) to be listed.
NOTE1: The wildcard character "*" (asterisk) may be used to specify a filter mask.

- "*file" indicates a suffix match, e.g. "*.bin", "/mvb/*.log"
  NOTE: The "*" must be the first character or the first character after the last directory separator (POSIX-style uses a slash "/", DOS/Windows-style uses a backslash "\" as directory separator).
- "file*" indicates a prefix match, e.g. "config*", "/mvb/config*"
  NOTE: The "*" must be the last character.

NOTE2: If operand "file" is omitted all entries of the current working directory will be listed.

EXAMPLES

List all entries of directory "/mvb" having the same prefix "config":

```
mvbMON>tnmfsls /mvb/config*
config1.bin
config2.bin
config3.bin
Total: 3 entries
mvbMON>
```

List all entries of directory "/mvb" (use POSIX-style long listing format):

```
mvbMON>tnmfsls -l /mvb (NOTE: same as "tnmfsls -l /mvb/*")
FileMode   #Lnk #UID #GID      Size LastModification  Name
drwxrwxrwx    1    0    0         0 Jun 17 2003 17:30 data
-rwxrwxrwx    1    0    0      4096 Sep  2 2002 10:37 config1.bin
-rwxrwxrwx    1    0    0      4096 Sep  5 2002 16:25 config2.bin
-rwxrwxrwx    1    0    0      4096 Jan  1 1970 00:00 config3.bin
-rwxrwxrwx    1    0    0    147456 Aug 21 2003 13:51 mvb_app.elf
Total: 5 entries (directories: 1, files: 4)
mvbMON>
```

NOTES:

The value of "FileMode" consist of:

- one character, which defines the entry type
- three permission fields of three characters each (owner permissions, group permissions, other permissions)

Entry type:

'-' regular file
'd' directory
'b' block special
'c' character special
'l' (lower-case ell) symbolic link
'p' FIFO

Each permission field has three character positions:

1. If 'r', the file is readable; if '-', the file is not readable.
2. If 'w', the file is writeable; if '-', the file is not writeable.
3. The first of the following that applies:
   - 'S' If in <owner permissions>, the file is not executable and set-user-ID mode is set. If in <group permissions>, the file is not executable and set-group-ID mode is set.
   - 's' If in <owner permissions>, the file is executable and set-user-ID mode is set. If in <group permissions>, the file is executable and set-group-ID mode is set.
   - 'T' If in <other permissions> and the file is a directory, search permission is not granted to others, and the restricted deletion flag is set.
   - 't' If in <other permissions> and the file is a directory, search permission is granted to others, and the restricted deletion flag is set.
   - 'x' The file is executable or the directory is searchable.
   - '-' None of the attributes of 'S', 's', 'T', 't' or 'x' applies.

List all entries of directory "/" (use file system specific long listing format):

```
mvbMON>tnmfsls -s /
FsType: TFS
FsAttr        Size Name
              4096 config1.bin
i             4096 config2.bin
              4096 config3.bin
e               68 monrc
BE          147456 mvb_app.elf
Total: 5 entries
mvbMON>
```

NOTES:

The value of "FsType" is a 32-bit hexadecimal number (e.g. 00000001). If the file system type is known, then a text is displayed instead of the number (e.g. 00000001=TFS).

The value of "FsAttr" is a 32-bit hexadecimal number (e.g. 80000200). If the meaning of the value is known, then a text is displayed instead of the number (e.g. bitset 00000014="BE"; bit 00000004="B"; bit 00000010="E").

If the TNM managed file system is of type "TFS" (Tiny File System; flash file system offered by boot loader "MicroMonitor" [7]; e.g. d113), then the value of "FsAttr" may contain a combination of the following file attributes:

- 'e' executable script file
- 'b' file is to be automatically run at boot time
- 'B' file is to be automatically run at boot time, after querying user
- 'l' (lower-case ell) symbolic link file
- 'E' executable binary file, i.e. ELF file format
- 'c' file is compressed
- 'i' file is in-place-modifiable
- 'u' file is not readable when monitor is below required user level
- '1' user level 1
- '2' user level 2
- '3' user level 3

REFERENCES

See TNM services "TNM_FS_opendir, TNM_FS_closedir, TNM_FS_readdir, TNM_FS_rewinddir, TNM_FS_stat" in d-000487, TCN Software Architecture [3] and in IEC 61375-2, Real-Time Protocols [6].
See also UNIX utility "ls" and POSIX procedures "opendir, closedir, readdir, rewinddir, stat" in "The Single UNIX Specification,. Version 3" [8].

tnmfsrm – remove directory entry

USAGE

tnmfsrm file

DESCRIPTION

The command removes the directory entry (i.e. file) specified by the "file" operand.

- The user must confirm the execution of the command ("continue?" query).

OPTIONS

-

OPERANDS

file

An absolute or relative pathname of a directory entry (i.e. file) to be removed.
NOTE: The wildcard character "*" (asterisk) may be used to specify a filter mask.

- "*file" indicates a suffix match, e.g. "*.bin", "/mvb/*.log"
  NOTE: The "*" must be the first character or the first character after the last directory separator (POSIX-style uses a slash "/", DOS/Windows-style uses a backslash "\" as directory separator).
- "file*" indicates a prefix match, e.g. "config*", "/mvb/config*"
  NOTE: The "*" must be the last character.

EXAMPLES

Remove file "config1.bin" from the current working directory:

```
mvbMON>tnmfsrm config1.bin
continue? [y/n] y
File: config1.bin deleted successfully !
mvbMON>
```

Remove file(s) "*.log" from directory "/mvb":

```
mvbMON>tnmfsrm /mvb/*.log
continue? [y/n]
…
mvbMON>
```

Remove all files from the current working directory:

```
mvbMON>tnmfsrm *
continue? [y/n]
…
mvbMON>
```

REFERENCES

See TNM service "TNM_FS_unlink" in d-000487, TCN Software Architecture [3] and in IEC 61375-2, Real-Time Protocols [6].
See also UNIX utility "rm" and POSIX procedure "unlink" in "The Single UNIX Specification,. Version 3" [8].

tnmfsxmodem – xmodem file transfer

USAGE

tnmfsxmodem [-d|-u][-ov][-cky][-f file][-s size][-m mode][-t type -a attr]

DESCRIPTION

The command initiates an Xmodem (Xmodem-CRC, Xmodem-1K, Ymodem) file transfer.

OPTIONS

-d

Download a file.

-u

Upload a file. The option "-f" must also be specified.

-o

Overwrite a file, if it still exist. The option "-d" must also be specified.

-v

Verify only. The option "d" must also be specified.

-c

Use CRC instead of checksum. Auto detected during upload (option "-u").

-k

Use 1K block size (instead of default 128 byte).
NOTE: May be used for upload (option "-u"). Auto detected during download (option "-d").

-y

Support the Ymodem extension to Xmodem. When "-y" is specified, "-c" and "-k" shall be assumed, "-f", "-s" and "-m" are not allowed.

-f file

For upload or download, specifies an absolute or relative pathname of the file to transfer.

-s size

Since Xmodem transfers in fixed block sizes (128 or 1024 bytes), the computed download size of a file is likely to be incorrect. This option allows the user to override the compute size with the value specified by "size".
NOTE: Required for Xmodem download on TFS file system.

-m mode

Set the file permission bits of the newly-created file to the specified "mode" value.
NOTE: May be used only for download.
The file permission bits are specified by a non-negative octal number. For each bit set in the octal number, the corresponding file permission bit shown in the following table will be set; all other file permission bits will be cleared.

Octal	Mode bit	Octal	Mode bit	Octal	Mode bit	Octal	Mode bit
4000	S_ISUID	0400	S_IRUSR	0040	S_IRGRP	0004	S_IROTH
2000	S_ISGID	0200	S_IWUSR	0020	S_IWGRP	0002	S_IWOTH
		0100	S_IXUSR	0010	S_IXGRP	0001	S_IXOTH
		0700	S_IRWXU	0070	S_IRWXG	0007	S_IRWXO

NOTE: Assume file mode 0777 (i.e. the value of the bitwise inclusive OR of S_IRWXU, S_IRWXG and S_IRWXO) if this option is omitted.

-t type

Type of the TNM managed file system (32-bit number or symbolic constant). The option "-a" must also be specified.
NOTE: May be used only for download.
Known symbolic file system types:

- TFS (tiny file system)

-a attr

File system specific file attributes (32-bit number or symbolic constant). The option "-t" must also be specified.
NOTE: May be used only for download.
Known symbolic file attributes for file system type "TFS":

'e' executable script file
'b' file is to be automatically run at boot time
'B' file is to be automatically run at boot time, after querying user
'E' executable binary file, i.e. ELF file format
'c' file is compressed
'u' file is not readable when monitor is below required user level
'1' user level 1
'2' user level 2
'3' user level 3

OPERANDS

-

EXAMPLES

Download the file "mvb_app.elf" to the current working directory by using Xmodem protocol. Specify the file name, the file size and the TFS file system specific file attributes:

```
mvbMON>tnmfsxmodem -d -f mvb_app.elf -s 147456 -t TFS -a E
…
```

Upload the file "/mvb/config1.bin" by using the Ymodem protocol:

```
mvbMON>tnmfsxmodem -u -y -f /mvb/config1.bin
…
```

REFERENCES

See TNM services "TNM_FS_open, TNM_FS_close, TNM_FS_read, TNM_FS_write" in d-000487, TCN Software Architecture [3] and in IEC 61375-2, Real-Time Protocols [6].
See also POSIX procedures "open, close, read, write " in "The Single UNIX Specification,. Version 3" [8].

Application Examples

Passive MVB Device

The MVB monitor device can be used to get information about the MVB network to which it is connected.

Step 1

Connect to MVB using input line A and B (by default the own device address is 0x000, which is typically not used in a MVB network):

```
mvbMON>svc -l ab
device address: 0x000 (0)
line config   : AB
mvbMON>
```

Step 2

Scan for an external MVB bus administrator (BA):

```
mvbMON>sva -S
Scan for BA (max. 16seconds).........
BA found (see details with command 'svd') !
mvbMON>
```

Step 3

List all devices (identified by its device address) and their device status found on the MVB network during devices scan of the external MVB bus administrator.
The list contains 3 devices:

- device 0x000 (itself, own device; see command "svc")
- device 0x002 (device which is running the MVB bus administrator, since bits BA=, CS2=1 and CS3=1)
- device 0x010 (MVBC class 1 device, since bits SP=1, BA=1, GW=1 and MD=1)

```
mvbMON>svd -a
number of devices: 3
-------------------------------------------------------------------
|              |        |         | C C C C | L R S S E F D S |
| device       | device | S B G M | S S S S | A L S D R R N E |
| address      | status | P A W D | 0 1 2 3 | T D D D D C R R |
-------------------------------------------------------------------
| 0x000 (   0) | 0x0080 | 0 0 0 0 | 0 0 0 0 | 1 0 0 0 0 0 0 0 |
| 0x002 (   2) | 0x5380 | 0 1 0 1 | 0 0 1 1 | 1 0 0 0 0 0 0 0 |
| 0x010 (  16) | 0xF0C0 | 1 1 1 1 | 0 0 0 0 | 1 1 0 0 0 0 0 0 |
-------------------------------------------------------------------
mvbMON>
```

Step 4

Scan for process data ports transmitted over the MVB and configure it as SINK (i.e. this allows to receive dataset values of the ports).

NOTE: It is not possible to detect the source device of the process data ports.

```
mvbMON>pc -S
Scan for ports.............
Scan result (2 ports configured as SINK):
0x002 (   2)  SINK    32 bytes (MVB F-code 4)
0x010 (  16)  SINK     2 bytes (MVB F-code 0)
number of ports: 2
Configure dynamic BA..........
mvbMON>
```

Step 5

Get dataset and its freshness timer from port 0x10. The first access returns a freshness timer of 784ms, the second access returns a freshness timer of 32ms. So it seems, that the MVB bus administrator uses a cycle time of 1024ms for that port.

The dataset values of the port did not change between both access.

NOTE: The dataset values depends on the application of the device, where the port is configured as SOURCE.

```
mvbMON>dsg 0x10
freshness timer:
----------------
784ms (0x0310)

dataset values:
---------------
0xC0 0x03
mvbMON>
mvbMON>dsg 0x10
freshness timer:
----------------
32ms (0x0020)

dataset values:
---------------
0xC0 0x03
mvbMON>
```

MVB Test Master Device

During development of a MVB device (e.g. door control, climate control,...) the MVB monitor device can be used for:

- check, if communication to the DUT (i.e. MVB Device Under Test) is possible
- check, if the device address of the DUT is configured as expected
- check, if the process data "source" ports of the DUT are configured as expected
- perform MVB process data communication with the DUT (i.e. send dataset to DUT, receive dataset from DUT)

Step 1

Connect to MVB using input line A and B (by default the own device address is 0x000, which is typically not used in a MVB network):

```
mvbMON>svc -l ab
device address: 0x000 (0)
line config   : AB
mvbMON>
```

Step 2

Enable the own MVB bus administrator (BA) in automatic mode:

```
mvbMON>sva auto
Configure dynamic BA..........
BA state: AUTO
mvbMON>
```

Step 3

Wait approximately 20seconds (i.e. time required for the own MVB bus administrator to perform devices scan over the whole device address range).

Step 4

List all devices (identified by its device address) and their device status found on the MVB network during devices scan of the own MVB bus administrator. The list contains 2 devices:

- device 0x000 (itself, own device; see command "svc"; device which is running the MVB bus administrator, since bits BA=1, CS2=1 and CS3=1)
- device 0x020 (DUT which is a MVB class 2 device, since bits SP=0, BA=0, GW=0 and MD=1)

```
mvbMON>svd -a
number of devices: 2
-------------------------------------------------------------
|              |        |         | C C C C | L R S S E F D S |
| device       | device | S B G M | S S S S | A L S D R R N E |
| address      | status | P A W D | 0 1 2 3 | T D D D D C R R |
-------------------------------------------------------------
| 0x000 (   0) | 0x0080 | 0 0 0 0 | 0 0 0 0 | 1 0 0 0 0 0 0 0 |
| 0x020 (  32) | 0x1080 | 0 0 0 1 | 0 0 0 0 | 1 0 0 0 0 0 0 0 |
-------------------------------------------------------------
mvbMON>
```

Step 5

Scan for process data ports transmitted over the MVB (i.e. from DUT) and configure it as SINK (i.e. this allows to receive dataset values of the ports).

NOTE: The MVB bus administrator will be reconfigured using the detected process data ports (with fastest possible cycle time).

```
mvbMON>pc -S
Scan for ports.................................................
..............
Scan result (3 ports configured as SINK):
0x010 (  16)  SINK     2 bytes (MVB F-code 0)
0x011 (  17)  SINK    32 bytes (MVB F-code 4)
0x030 (  48)  SINK     8 bytes (MVB F-code 2)
number of ports: 3
Configure dynamic BA..........
mvbMON>
```

Step 6

Configure process source data ports, which should be used to transmit dataset values to the DUT:

- port 0x200 as SOURCE with port size 32 bytes (MVB F-code 4)
- port 0x300 as SOURCE with port size 8 bytes (MVB F-code 2)

NOTE: The MVB bus administrator will be reconfigured using the specified process data ports (with fastest possible cycle time).

```
mvbMON>pc -t source -s 32 0x200
Configure dynamic BA..........
mvbMON>
mvbMON>pc -t source -s 8 0x300
Configure dynamic BA..........
mvbMON>
```

Step 7

Perform MVB process data communication (example):

- receive dataset from port 0x010
- transmit dataset to port 0x200

```
mvbMON>dsg 0x10
freshness timer:
----------------
0ms (0x0000)

dataset values:
---------------
0x80 0x00
mvbMON>
mvbMON>dsp -2 0x200 0xcafe
mvbMON>
mvbMON>dsg 0x200
freshness timer:
----------------
65535ms (0xFFFF)

dataset values:
---------------
0xCA 0xFE 0x00 0x00 0x00 0x00 0x00 0x00
0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00
0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00
0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00
mvbMON>
```

MVB Bus Administrator Device

The MVB monitor device can be used as "normal" MVB bus administrator bus node.

Step 1

Set own device address to 0x001 and connect to MVB using input line A and B:

```
mvbMON>svc -a 1 -l ab
device address: 0x001 (1)
line config   : AB
mvbMON>
```

Step 2

Get MVB bus administrator configuration (i.e. BA object) from file "ba_tcn.bin" and start the own MVB bus administrator (BA).

NOTE1: The file "ba_tcn.bin" must be located in the file system of the MVB monitor device.
NOTE2: The file "ba_tcn.bin" must have the binary structure of "MVB_Administrator" which is defined in IEC 61375-3 [4].

```
mvbMON>sva tcn ba_tcn.bin
BA state: TCN (ba_tcn.bin)
mvbMON>
```

MVB Device defined by D2000 Configuration Tool

The MVB monitor device can be configured as a "normal" bus node using the duagon D2000 configuration tool [2] .

NOTE1: The file "dev_d113.bin" must be located in the file system of the MVB monitor device.
NOTE2: The file "dev_d113.bin" must be generated by the D2000 configuration tool

```
mvbMON>cm d2000 dev_d113.bin
Parse configuration file.......
Parse configuration file.......

configuration status - device:
------------------------------
device address: 0x020 (32)
line config   : AB

configuration status - ports:
-----------------------------
number of ports: 8
0x002 (   2)  SINK    32 bytes (MVB F-code 4)
0x01E (  30)  SINK     4 bytes (MVB F-code 1)
0x123 ( 291)  SINK     2 bytes (MVB F-code 0)
0x456 (1110)  SOURCE   4 bytes (MVB F-code 1)
0x789 (1929)  SOURCE   8 bytes (MVB F-code 2)
0xABC (2748)  SOURCE  32 bytes (MVB F-code 4)
0xCBA (3258)  SOURCE  16 bytes (MVB F-code 3)
0xDEF (3567)  SINK    32 bytes (MVB F-code 4)
number of ports: 8

configuration status - bus administrator:
-----------------------------------------
BA state: D2000 (dev_d113.bin)

mvbMON>
```

Download a file over MVB

The MVB monitor device can be used as TNM manager to download a file over MVB to the file system of a remote station.

Step 1

Set own device address to 0x001 and connect to MVB using input line A and B:

```
mvbMON>svc -a 1 -l ab
device address: 0x001 (1)
line config   : AB
mvbMON>
```

Step 2

If no bus administrator (BA) is connected to the MVB, then enable the own bus administrator in automatic mode:

```
mvbMON>sva auto
Configure dynamic BA..........
BA state: AUTO
mvbMON>
```

Step 3

Select the remote station by defining of its network address:

```
mvbMON>tnm -s 7
node   : 0x00 (  0) SAME_NODE
station: 0x07 (  7)
topo   : 0x00 (  0) ANY_TOPO
mvbMON>
```

Step 4

Reserve station for exclusive use of this manager (time out=300 seconds=5 minutes):

```
mvbMON>tnmwsr -t 300 1
mvbMON>
```

Step 5

List all entries of the current working directory:

```
mvbMON>tnmfsls
app.elf
monrc
pld.bin
Total: 3 entries
mvbMON>
```

Step 6

Download the file "config.bin" by using Xmodem to the current working directory:

```
mvbMON>tnmfsxmodem -d -f config.bin -s 4096
…
```

Step 7

List all entries of the current working directory:

```
mvbMON>tnmfsls
app.elf
config.bin
monrc
pld.bin
Total: 3 entries
mvbMON>
```

Appendix

Appendix A: Document History

d-000551-002683

- Official release of TNM (Train Network Management).
- Multiple minor changes, mainly editorial.

d-000551-002458

- NOTE: Only for duagon internal use.
- add TNM commands
- modify comment of option "-a" in command "svc" (Devices with MVB message data capability must use device address range 0x01…0xFE; not 0x00…0xFF).

d-000551-001951

- created

Index

?

A

B

C

D

E

F

H

I

L

M

N

P

R

S

T

V

W

X

Y

Reference

1 Duagon GmbH, Badenerstrasse 21, CH-8953 Dietikon,
Phone: +41-44-743 40 00,
Fax: +41-44-743 40 15,
Internet: http:\\www.duagon.com

2 D2000 Configuration Tool – User's Guide;
Document Nr.: d-000341;
Revision: d-000341-002677, October 11th 2004, J. Rapka

3 TCN Software Architecture – Data Sheet;
Document Nr.: d-000487;
Revision: d-000487-002679, November 18th 2004, J. Rapka

4 IEC (International Electrotechnical Commission);
Electric Railway Equipment – Train Bus –
Part 1: Train Communication Network;
Clause 3: Multifunction Vehicle Bus;
Document No.: IEC 61375-1;
Document date: 1999-09
Document status: First edition

5 IEC (International Electrotechnical Commission);
Electric Railway Equipment – Train Bus –
Part 1: Train Communication Network;
Clause 5: Train Network Management;
Document No.: IEC 61375-1;
Document date: 1999-09
Document status: First edition

6 IEC (International Electrotechnical Commission);
Electric Railway Equipment – Train Bus –
Part 1: Train Communication Network;
Clause 2: Real-Time Protocols;
Document No.: IEC 61375-1;
Document date: 1999-09
Document status: First edition

7 Boot Loader "MicroMonitor" - Data Sheet;
Document Nr.: d-000552;
Revision: d-000552-002676; October 12th 2004, Rapka

8 The Singe UNIX Specification, Version 3;
Copyright © 2001-2004 The IEEE and The Open Group,
All Rights Reserved.