NAT-MCH-Gen4
CLI and Script
User Guide V1.1

N.A.T. Gesellschaft für Netzwerk- und Automatisierungs-Technologie mbH
Konrad-Zuse-Platz 9 | 53227 Bonn, Germany | Phone: +49 228 965 864 - 0
sales@nateurope.com | www.nateurope.com

Disclaimer

The following document, compiled by N.A.T. GmbH (henceforth called N.A.T.), represents the current status of the product´s development. The document is updated on a regular basis. Any changes which might ensue, including those necessitated by updated specifications, are considered in the latest version of this documentation. N.A.T. is under no obligation to notify any person, organization, or institution of such changes or to make these changes public in any other way.

This document has been compiled with great care, however the document could still include technical inaccuracies or typographical errors.

N.A.T. offers no warranty, either expressed or implied, for the contents of this documentation or for the product described therein, including but not limited to the warranties of merchantability or the fitness of the product for any specific purpose.

In no event will N.A.T. be liable for any loss of data or for errors in data utilization or processing resulting from the use of this product or the documentation. In particular, N.A.T. will not be responsible for any direct or indirect damages (including lost profits, lost savings, delays or interruptions in the flow of business activities, including but not limited to, special, incidental, consequential, or other similar damages) arising out of the use of or inability to use this product or the associated documentation, even if N.A.T. or any authorized N.A.T. representative has been advised of the possibility of such damages.

The use of registered names, trademarks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations (patent laws, trade mark laws, etc.) and therefore free for general use. In no case does N.A.T. guarantee that the information given in this document is free of such third-party rights.

Neither this document nor any part thereof may be copied, translated, or reduced to any electronic medium or machine form without the prior written consent from N.A.T. GmbH.

This product (and the associated documentation) is governed by the N.A.T. General Conditions and Terms of Delivery and Payment.

Note:

The release of the User Manual is related to a certain HW board revision.
For HW revisions earlier than the one given chapter “Supported HW Revisions“ please contact N.A.T. for the corresponding older Manual release.

Table of Contents

List of Tables

List of Figures

Es konnten keine Einträge für ein Abbildungsverzeichnis gefunden werden.

Conventions

If not specified otherwise, addresses and memory maps are provided in hexadecimal notation, identified by *0x*. Table 1 lists the abbreviations used in this document:

Table 1: List of used abbreviations

Abbreviation	Description
AMC	Advanced Mezzanine Card
ATCA	Advanced Telecommunications Computing Architecture
CRC	Cyclic Redundancy Check
DIP SW	Dual In-Line Switch
EEPROM	Electrically Erasable PROM
FPGA	Field Programmable Gate Array
GbE	Gigabit Ethernet
HS	Hot Swap
I^2C	Inter-Integrated Circuit
I/O	Input/Output
IP	Internet Protocol
IPMB	Intelligent Platform Management Bus
IPMI	Intelligent Platform Management Interface
JTAG	Joint Test Action Group
µC	Microcontroller
µTCA	Micro Telecommunications Computing Architecture
MUX	Multiplexer
PCB	Printed Circuit Board
PCI(e)	Peripheral Component Interconnect (Express)
Rx	Receiver
RAM	Random Access Memory
(P)ROM	(Programmable) Read Only Memory
PLL	Phase Locked Loop
SFP	Small Form-Factor Pluggable
TCKL	Telecom Clock

1 INTRODUCTION

The NAT-MCH-G4 CLI provides a modern Linux style Command Line Interface for monitoring and configuring the MCH via a serial console type interface.

This manual provides a short introduction to the CLI, the associated script language and explains its basic functionality.

The CII can be used via the following interfaces:

- *USB-Console*
- *Telnet*
- *Live Log of the web interface*

Compared to the LCI of the 3rd generation NAT-MCH, the new CLI of the 4th generation NAT-MCH-G4 provides:

- *A reduced number of first level key words*
- *A help functionality at any command level*
- *A flexible and adaptable support of parameters and options*

1.1 BASIC COMMANDS:

The syntax of all CLI commands is as follows:

command [parameter 1] … [parameter n] -[option 1] … -[option n]

The following basic commands are supported:

set - manipulates the value described in parameter section

get – retrieves the value described in parameter section

print – prints information and/or values described in parameter section

help – the basic CLI help command

ifconfig - TCP/IP interface configuration

script – script support

ping – pings another network node in the network

2 USING THE CLI

A good starting point for using the CLI always is invoking the <help> command at the CLI prompt:

nat> help

The <help> command will display all commands with their respective parameters.

More detailed information abaout a specific function can be obtained by the sub level help system by adding the <-help> option:

Example:

nat> set fan level -help

Output:

usage: set fan level [<params>]
params:
[<site_number>]
[<dev_id>]
<level>

Set fan speed level of cooling units

2.1 USAGE OF OPTIONS AND PARAMETERS:

The CLI supports mandatory and optional parameters.

Optional parameters are shown in square brackets, i.e.: [site_number]

Options must be placed at the end of the command string:

Example:

Nat> set debug ipmi min -p

Sets the debug level for the IPMI interface to minimum and **p**ersistently stores the value.

Persistently stored values are safe against power cycles and re-boots.

2.2 DEBUGGING AND LOGGING

All N.A.T. software and hardware products are parsed through an intensive qualification process. Nevertheless, issues can arise especially in a multivendor environment. In this case N.A.T. might ask you to provide certain debug information from the running system in order to help accelerating the issue analysis. In most cases this will be a log file. The system constantly generates debug information and logs errors into the ***syslog*** buffer.

By default, debug messages are not displayed at the console !

The ***syslog buffer*** is a power save circular buffer of 128Kbyte. Its contents can be shown by:

nat> print syslog buffer <opt>

The print syslog buffer command provides the options:

-c = clears the complete buffer after printing
-f = follow tail - prints any new entry in the syslog to the console

In some cases, further debug information might be required. Depending on the issues to investigate on, certain debug information can be created by setting debug levels.

Example:

nat> set debug ipmi min 6 -p

This command enables IPMI message logging for all messages exchanged with FRU 6 (AMC2). The flag -p makes the command persistent, i.e. you can reboot or power cycle the system and it will come up again with this setting. Please consult the "set debug -help" for more information about the debug options.

Debug flags can be set and cleared by the console of the WEB interface as well.

2.3 FURTHER DOCUMENTATION

This CLI manual is supposed to only provide a short introduction to and overview of the operation of the CLI.

If you are looking for a more specific description of certain sets of commands for certain hardware submodules of the NAT-MCH-G4, i.e. how to configure the clock or hub modules, please refer to the user manual for the respective submodule.

3 MCH GEN4 SCRIPT SUPPORT

The script support of the MCH uses the available CLI commands. Thus, a script consists of several CLI commands and can be edited and stored. The configuration scripts are divided into so-called domains which can be individually edited/stored/deleted. Any individual script is loaded during the start-up phase of the MCH, respectively the start-up of its submodules. i.e. the clock configuration is loaded when the clock module is recognized and initialized.

Table 2: Configuration Domains

Domain	Description
system	basic system configuration: executed during start-up
user	user space configuration: executing immediately after start-up
base1	Base configuration: Ethernet switch initialization of the Base-MCH (side 1)
base2	Base configuration: Ethernet switch initialization of the Base-MCH (side 2)
hub1	Switch configuration: Ethernet switch initialization of the hub module (side 1)
hub2	Switch configuration: Ethernet switch initialization of the hub module (side 2)
clk1	Clock configuration: Initialization of the clock module (side 1)
clk2	Clock configuration: Initialization of the clock module (side 2)

3.1 SCRIPT COMMANDS

The MCH supports the following CLI commands to create and clear a script configuration:

- script start <domain>
- script end
- script clear <domain | all>

The console interface can be used to enter a new script line-by-line. It is "best practice" Bu to create/edit the script offline and then download it to the MCH via the web-interface.

3.2 SHOW CONFIGURATION/AGGREGATED CONFIGURATION

The MCH supports the CLI command "print config <domain>" to display the possible configurations available for this particular domain:

Example:

nat> print config base1

Output:

set switchport state AMC1/0 ena
set switchport state AMC2/0 ena
set switchport state U1 ena
set switchport state CPU_1 ena

Use “print config all” instead of the ““print config <domain>” to show the *aggregated configurations* of the MCH:

Nat> print config all

Output:

script start system
ifconfig te0 192.168.1.41 255.255.255.0 0.0.0.0
set ekey ignore false xaui
script end

script start user
script end

script start base1
set switchport state AMC1/0 ena
set switchport state AMC2/0 ena
set switchport state U1 ena
set switchport state CPU_1 ena
script end

script start base2
script end

script start hub1
script end

script start hub2
script end

script start clk1
script end

script start clk2
script end

The *aggregated configuration* consists of all available configurations, concatenated by special *script commands*. It is “best practice” again to use the aggregated configuration for configuring your MCH line-by-line and to store the respective configuration.

3.3 SCRIPT COMMANDS

The MCH supports the following CLI commands to manage the script configuration:

- script start <domain>
- script end
- script clear <domain | all>

3.4 Creating a New Configuration

Creating a new script configuration is done in three steps:

Step 1 - Open a script configuration for writing: Please, enter “script start <domain>” to open appropriated script for writing mode.

Step 2 - add command to a script: All following commands will be added to <domain>.

NOTE: The commands are only in the script only, they are not executed at this point. The script configuration is applied after the next restart only!

Step 3 - close and store the script: Please use CLI command “script end” to close a script configuration for writing and to store it in non-volatile memory.

3.5 Default Configuration

A valid configuration is required for proper Ethernet switch operation. If no persistent configuration is available, the switch ports cannot enter an operational state.

An MCH in its factory-default state, or an MCH whose configuration has been deleted, does not contain a persistent switch configuration. In such cases, a default configuration is automatically generated and applied during system startup.

The default configuration ensures that all switch ports are initialized into a defined state. Without this configuration, all ports remain unconfigured or disabled.

Because the default configuration is generated and applied during MCH startup, it can be displayed using the following CLI command:

print config <domain>

Example:

```
script start base2
set switchport state   base2 AMC1/1 dis
set switchport state   base2 AMC2/1 dis
set switchport state   base2 AMC3/1 dis
set switchport state   base2 AMC4/1 dis
set switchport state   base2 AMC5/1 dis
set switchport state   base2 AMC6/1 dis
set switchport state   base2 AMC7/1 dis
set switchport state   base2 AMC8/1 dis
set switchport state   base2 AMC9/1 dis
set switchport state   base2 AMC10/1 dis
set switchport state   base2 AMC11/1 dis
set switchport state   base2 AMC12/1 dis
set switchport state   base2 U1 dis -s
set switchport state   base2 U1A dis -s
set switchport state   base2 U1B dis -s
set switchport state   base2 U2 dis -s
set switchport state   base2 U2A dis -s
set switchport state   base2 U2B dis -s
set switchport state   base2 UPDC_B dis
set switchport state   base2 ISWC_B dis
set switchport state   base2 CPU_1 ena
script end
```

The default configuration is volatile. It is not stored persistently and is regenerated and reapplied at each system startup.

To store the currently active default configuration permanently in MCH FLASH, use the following CLI command: ***script config save***

NOTE: If a user-defined configuration is required, it is recommended to use the default configuration as the baseline.

3.6 SUBSHELL CPSS

The CPSS subshell is a command shell integrated into the MCH CLI. It provides a gateway to the native CLI of the Marvell Ethernet switches and enables the direct invocation of CPSS functions.

WARNING: Commands executed within the CPSS subshell are not part of the regular switch management layer and are therefore not subject to its abstraction and validation mechanisms. Use of the CPSS subshell **requires in-depth knowledge** of the underlying switch architecture and the CPSS interfaces! Improper use may result in unexpected behaviour or switch misconfiguration.

The CPSS subshell is started with the following command:
cpss <domain {hub1|hub2}>

The subshell is exited with:
exit

Example - Using the CPSS Subshell

```
nat> cpss hub2
cpss> do cpss-api call cpssPxSinglePortModeSpeedSet devNum 2 portNum 0 powerUp true ifMode KR speed 10000
cpss> do cpss-api call cpssPxSinglePortModeSpeedSet devNum 2 portNum 4 powerUp true ifMode KR speed 10000
cpss> do cpss-api call cpssPxSinglePortModeSpeedSet devNum 2 portNum 8 powerUp true ifMode KR speed 10000
cpss> exit
```

CPSS subshell commands can also be used within a configuration.

Example - CPSS Subshell in a Configuration

```
script start hub2
set switchport state hub2 AMC1/8-11 dis
set switchport state hub2 AMC2/8-11 ekey
set switchport state hub2 AMC3/8-11 dis
cpss hub2
do cpss-api call cpssPxSinglePortModeSpeedSet devNum 2 portNum 0 powerUp true ifMode KR speed 10000
do cpss-api call cpssPxSinglePortModeSpeedSet devNum 2 portNum 4 powerUp true ifMode KR speed 10000
do cpss-api call cpssPxSinglePortModeSpeedSet devNum 2 portNum 8 powerUp true ifMode KR speed 10000
exit
script end
```

Appendix A: Known issues

Currently there are no known issues with firmware 3.1.1.

Appendix B: Document's History

Revision	Date	Description	Author
1.0	10.08.25	initial release	hl
1.1	11.05.2026	Added chapter "3.5 Default Configuration"	al
	11.05.2026	Added chapter "3.6 SUBSHELL CPSS"	al
	14.07.2026	Added debugging and logging chapter	hl