User's Guide

NV1600 Flashtec™ NVRAM Drives Command Line Utility

Released

July 2018

Microsemi

a Microchip company

Contents

Tables

1 Revision History

Revision	Revision Date	Details of Change
5	July 2018	Issue 5 of this document includes updates for the v1.7 maintenance release for NV1616 only. **Card-Level Commands** • Added the Card-Level command Get Backup Results. Removed copyright information section and updated branding. Note that the changes for the v1.5 maintenance release also apply to NV1616 only.
4	November 2017	Issue 4 of this document includes updates for the v1.5 maintenance release. **NVRAM-Level Commands** Added the following NVRAM-Level commands: • Check Hang Card • Reset Hang Card **Card-Level Commands** • Added event type NVRAM_EVENT_TYPE_UC_ERROR to Register Event Handler. **Debug-Level Commands** • Added ddr set to to Register Access.
3	October 2016	V1.4 Maintenance release.
2	May 2016	V1.3 Maintenance release.
1	October 2015	Document created.

1 About this Guide

This guide describes the command line utility provided with Microsemi's Flashtec™ NVRAM Drive software. It focuses on using the interactive features of the CLI and provides descriptions of the CLI commands.

1.1 About the Utility

The Flashtec NVRAM Drive software includes a command line utility, or CLI, called *pmcnvm*. You can use the CLI as an administration tool to explore and configure the Flashtec NVRAM Drive interactively without issuing API calls from a host application.

The CLI is a demonstration program that provides a "wrapper" around the Flashtec NVRAM Drive API library. It is distributed in source format so that you can use it as an example for implementing the API functions and linking the API library in your own application.

1.2 How to Find More Information

You can find more information about using the NVRAM drive and its APIs by referring to these documents, available for download at www.pmcs.com/myPMC:

1. *NV1600 Flashtec™ NVRAM Drives Installation and User's Guide*—Describes how to set up the NV1600 Flashtec NVRAM Drive, install drivers and API libraries. (PMC-2142284)
2. *NV1600 Flashtec™ NVRAM Drives Programmer's Manual*—Describes the API functions, events, structures, and data types for developing NV1600 NVRAM Drive host applications. (PMC-2150885)
3. *NV1600 Flashtec™ NVRAM Drives Firmware Release Notes*—Provides updated driver and firmware information, usage notes, and known issues. Included with the software/firmware release package. (PMC-2150854)

2 Using the Utility

Note: Be sure you complete the installation of the Flashtec NVRAM Drive software before attempting to launch the CLI program. See "Installing the Flashtec NVRAM Drive Software" in the *NV1600 Flashtec NVRAM Drives Installation and User's Guide* [1].

2.1 Building the Command Line Utility

See the *NV1600 Flashtec Installation and User's Guide* [1] for detailed instructions for installing and building the CLI, and the prerequisite software:

- Compiling and installing the kernel modules
- Loading and testing the NVMe driver
- Building and compiling the API library
- Building the CLI

Note: To build the CLI program, you must have superuser (root) privilege.

2.2 Starting the Command Line Utility

Note: To run the CLI program, you must have superuser (root) privilege.

1. Launch the CLI program from the Linux shell. In this example, the installation directory is `<release_root>/demo`.

```
$ su - root
# <release_root>/demo/pmcnvm
```

2. At the in-program prompt, enter the NVRAM level to start using the CLI:

```
PMCNVM >nvram
```

After starting the CLI, you initialize the card and then perform other tasks. See the Usage Example on page 46 for reference.

CLI organization and workflow is described in the next sections.

2.3 How the Command Line Utility is Organized

The CLI is organized as a series of levels or directories. At each level, you can run only certain commands. In some levels, the CLI imposes a strict order where you can only complete commands (or tasks) that are prerequisites for commands (or tasks) at other levels. You must complete the prerequisite commands before you can proceed to the next level. In other levels, you can move freely to lower-level directories.

For example, the top-most level of the CLI is the "nvram" directory. At this level, you can list the Flashtec NVRAM Drives installed on your machine (the "card list"), initialize one of the available cards, and select a card. Selecting a card moves you to the next level or directory, at which point, subsequent commands apply only to the selected card. (You don't have to specify the card again as long as you remain at that level.) However, you cannot move from the "nvram" directory to the "card" directory before you initialize the card. Upon entering the "card" directory, you can move to other directory levels ("ramdisk" or "mem", for instance) without completing any commands.

As you move from level-to-level, the CLI displays a list of available commands. For example, upon entering the "nvram" directory, the CLI displays the following command synopsis:

```
PMCNVM >nvram
card_list        card_list - gets list of the NVRAM cards in the system
init             init <card_idx> - establishes a connection with the NVM controller
                  and Initializes the NVRAM
```

```
release          release <card_idx> - closes the connection with NVM controller and
                  release system resources
show_cards       show_cards - show all the NVRAM cards in the system
card             card <card_idx> - Enter the specific card dir
```

The following table describes the main CLI directory levels. For a description of commands at each level, see CLI Commands on page 10.

Table 1 • CLI Directory Levels

Directory Level	Purpose
nvram	Show card list, initialize card, and select card
card	Card information, status, authentication, backup/restore options, statistics, and so on
mem	Memory mapping, read/write/reset memory, and dump memory to file
ramdisk	Format ramdisk, mount/unmount filesystem, copy/delete/list ramdisk contents
debug	Manage log files, read/dump DDR memory contents, get firmware version **Note:** By default, the debug commands are disabled. To enable the debug commands, authenticate the card first; see the section "Setting the Master and Admin Authentication Keys" in the *NV1600 Flashtec NVRAM Drives Installation and User's Guide* [1].

2.4 Navigating the CLI and Getting Help

The Command Line Utility conforms to Linux/Unix standards for navigation and getting help:

- To display help, type "?"
- To see a list of all commands available in the current level, type "ls"
- To browse the commands available in the current level, use the Tab key
- Use the Up Arrow and Down Arrow keys to display command history
- Type ".." to return to the previous level
- Type "..." to return to the root (top-most) level
- Type "exit" (from any level) to close the application and return to the Linux shell
- Use CTRL-H as an alternative to the Backspace key

  Note: Some Linux distributions may not enable the Backspace key, by default. Check your Linux Profile Preferences and ensure that "Backspace key generates CTRL-H" is enabled.

2.5 Auto-Completion

The Command Line Utility implements auto-completion for commands and parameters, similar to the Linux/Unix shell:

- Press the TAB key to complete the command
- Press the TAB key after white space to display candidate command parameters (one after the next)

For example, to display information about the ramdisk, the full command is:

```
PMCNVM/NVRAM/CARD 1 >info_get INFO_GROUP_RAMDISK
```

However, it might be faster to type:

```
PMCNVM/NVRAM/CARD 1 >inf<Tab><Space><Tab><Tab><Tab><Return>
```

This sequence auto-completes the `info_get` command, then it selects the third valid value for the command parameter: the RAM disk information group.

3 CLI Commands

The table below summarizes the CLI commands by directory level and category. For more information about CLI commands and parameters, see the subsequent sections.

Command	Summary	Reference
Directory Level: root		
nvram	Enters the NVRAM command level.	Enter NVRAM Level on page 13
Directory Level: nvram		
system_config_set	Sets the NVRAM system global configuration.	System Configuration Set
system_config_get	Gets the NVRAM system global configuration.	System Configuration Get
card_list	Shows list of NVRAM Drive cards in the system.	Get Card List on page 13
init	Initializes the NVRAM Drive and establishes a connection with the NVMe controller with a specified index.	Init Card on page 14
release	Closes the connection with NVMe controller with a specified index and releases system resources.	Release Card on page 14
show_cards	Shows all NVRAM Drive cards in the system.	Show Cards on page 14
card	Changes to a specific card commands directory level.	Navigate to Specific Card Level on page 15
Directory Level: card		
info_get	Retrieves card information by group.	Card Info Get on page 15
status_get	Retrieves dynamic card status by group.	Card Status Get on page 16
master_authenticate	Runs Master authentication between the host and NVMe card.	Master Authenticate on page 17
admin_authenticate	Runs Admin authentication between the host and NVMe card.	Admin Authenticate on page 17
set_config	Sets the NVRAM Drive card configuration.	Configuration Set on page 17
get_config	Gets the NVRAM Drive card configuration.	Configuration Get on page 21
bad_block_scan	Starts a bad block scanning operation on a flash bank. When finished, the event is received.	Scan Flash Bad Blocks on page 21
erase	Starts erasing a flash bank. When finished, the event is received.	Erase Flash Bank on page 21
backup	Starts backup of RAM contents to a flash bank. When finished, the event is received.	Backup on page 22
restore	Restores RAM contents from a flash bank. When finished, the event is received.	Restore on page 23

Command	Summary	Reference
bank_info	Retrieves information about a flash bank.	Flash Bank Info on page 23
vbb_count_get	Retrieves vendor bad block count for a specified block.	VBB (Vendor Bad Block) Count Get on page 23
restore_results_get	Returns information about the last restore operation that was run.	Restore Results Get on page 24
reg_ev_handler	Registers/unregisters the default handler for an event type.	Register Event Handler on page 24
time_of_day_sync	Synchronizes the time of day for the card to the current host time.	Time of Day Sync on page 25
statistics_get	Gets statistics from the NVRAM Drive card according to group.	Card Statistics Get on page 25
statistics_reset	Resets statistics from the NVRAM Drive card according to group.	Card Statistics Reset on page 26
sbl_download	Downloads the SBL to the card.	Download SBL on page 26
firmware_download	Downloads a firmware file to the card.	Download Firmware Image on page 27
fw_metadata_get	Retrieves firmware metadata information.	Firmware Metadata Get on page 27
heart_beat	Starts/stops the heartbeat transmission from the host to the NVRAM Drive card.	Heartbeat Management on page 28
reboot	Reboots the card.	Reboot the Card on page 28
pf_info	Reads Product Feature Info data.	Read PFI Data on page 28
lba_to_addr_map_get	Gets the DDR address of a specific LBA number when RAMDISK/DMI overlap is enabled.	Get the LBA to DDR Address on page 29
addr_to_lba_map_get	Gets the LBA number of a specific DDR address when RAMDISK/DMI overlap is enabled.	Get the DDR Address to LBA on page 29
ramdisk_enc_key	Sets the RAM disk encryption keys.	Set the RAM Disk Encryption Key on page 30
restore_corrupted	Starts restoring the image of the RAM content from a corrupted flash bank.	Restore Corrupted on page 30
self_test	Starts a self-test of the NVRAM Drive card.	Self Test on page 30
mem	Changes to the Memory commands directory level.	Enter the Memory Level on page 31
ramdisk	Changes to the RAM disk commands directory level.	Enter the RAM Disk Level on page 31
debug	Changes to the Debug commands directory level.	Enter the Debug Level on page 31
Directory Level: mem		

Command	Summary	Reference
memmap	Maps the NVRAM memory into host virtual memory. Other memory commands are allowed only after issuing this command.	Memory Map on page 32
write_mem	Writes the pattern to the mapped DMI.	Write to Memory on page 32
read_mem	Reads the mapped DMI.	Read from Memory on page 32
reset_whole_mem	Resets whole memory of the card.	Reset Whole Memory on page 33
set_mem_from_file	Writes data to DDR from a file.	Set Memory From File on page 33
dump_mem_to_file	Reads from card memory to a file.	Dump Memory To File on page 34
compare_mem_to_file	Compares card memory to the contents of a file.	Compare Memory to File on page 34
Directory Level: ramdisk		
format	Formats the RAM disk, using ext4 (similar to Linux command).	Format RAM Disk on page 35
mount	Mounts the RAM disk file system (similar to Linux command).	Mount the RAM Disk on page 35
unmount	Unmounts the RAM disk file system (similar to Linux command).	Unmount the RAM Disk on page 35
copy	Copies the file to the RAM disk *n* times.	Copy File to RAM Disk on page 36
delete	Removes (deletes) file from the RAM disk.	Delete File from RAM Disk on page 36
list	Shows the contents of the RAM disk.	List RAM Disk Files on page 36
Directory Level: debug		
print_log	Prints the log by page ID, or *node*.	Print Log on page 37
print_backup_log	Prints the backup log by page ID, or *node*.	Print Backup Log on page 38
log_polling	Enables/disables output of all log pages.	Log Polling on page 38
read_ddr	Reads from card DDR memory.	Read DDR on page 38
get_frim	Displays the mapping of bad blocks map per LUN/all.	Get FRIM Mapping on page 39
clear_frim_mapping	Clears the FRIM block mapping table.	Clear FRIM Mapping on page 39
register	Reads/writes to the controller CSR registers. Writes to DDR.	Register Access on page 39
dump_ddr_to_file	Copies DDR content to a file.	Dump DDR to File on page 40
fw_fail	Causes firmware to fail. For debug purposes only.	Test Firmware Fail on page 40
dump_block_to_file	Dumps specified flash block contents to a file.	Dump Flash Block to File on page 41
dump_page_to_file	Dumps specified flash page contents to a file.	Dump Flash Page to File on page 41

Command	Summary	Reference
inject_UC_errors	Injects uncorrectable errors while reading from flash.	Inject Uncorrectable Errors on page 42
bad_block_count_set	Sets the number of flash blocks as "bad" or "unmapped".	Set the Number of Bad Blocks on page 42
bad_block_set_by_lun	Set number of blocks in a specific LUN as “bad” or “unmapped".	Set the Number of Bad Blocks by LUN on page 43
write_to_block_from_file	Reads block contents from a file and writes it to a specific flash block.	Write to Flash Block from File on page 43
debug_config_set	Sets the NVRAM debug configuration.	Debug Configuration Set on page 44
debug_config_get	Gets the NVRAM debug configuration.	Debug Configuration Get on page 44
wait_for_event	Will cause the CLI to wait for event before returning, or until a 5 minute timeout occurs.	Wait For Event on page 45

3.1 Root-Level Commands

The root level displays the prompt, "`PMCNVM >`".

3.1.1 Enter NVRAM Level

Prototype

```
nvram
```

Description

This command enters the NVRAM command level.

Parameters

None

3.2 NVRAM-Level Commands

The NVRAM level displays the prompt, "`PMCNVM/NVRAM >`".

3.2.1 Get Card List

Prototype

```
card_list
```

Description

This command gets a list of the NVRAM cards in the system.

Note:
This command is a prerequisite to most other commands and should be executed first.

Parameters

None

3.2.2 Init Card

Prototype

```
init <card_idx>
```

Description

This command establishes a connection with the NVRAM card and initializes the NVRAM card with a specified index. This command must be performed prior to any other CLI command.

Parameters

Parameter	Type	Description	Values
card_idx	Mandatory	Index of the NVRAM card	0 - 16

3.2.3 Release Card

Prototype

```
release <card_idx>
```

Description

This command closes the connection with the NVRAM card with a specified index and releases system resources.

Parameters

Parameter	Type	Description	Values
card_idx	Mandatory	Index of the NVRAM card	0 - 16

3.2.4 Show Cards

Prototype

```
show_cards
```

Description

This command shows all the NVRAM cards in the system.

Parameters

None

3.2.5 Navigate to Specific Card Level

Prototype

```
card  <card_idx>
```

Description

This command navigates to a specific card command level.

Parameters

Parameter	Type	Description	Values
card_idx	Mandatory	Index of the NVRAM card	0 - 16

Note:
The card must be initialized (see Init Card on page 14) prior to this command.

3.3 Card-Level Commands

The card-level commands display the prompt, "PMCNVM/NVRAM/CARD # >".

3.3.1 Card Info Get

Prototype

```
info_get <group>
```

Description

This command retrieves general information about the NVRAM card by group.

Parameters

Parameter	Type	Description	Values
group	Mandatory	ID of info group	For general information, see the following table.. For detailed descriptions of the groups, see the section "NVRAM Info Group" in the *NV1600 Flashtec™ NVRAM Drive Programmer's Manual* [2]

Output

Info values for each group.

Table 2 • NVRAM Info Group Levels

Field	Description
INFO_GROUP_GENERAL	NVRAM General Information Group
INFO_GROUP_DMI	NVRAM DMI Information Group

Field	Description
INFO_GROUP_RAMDISK	NVRAM RAM Disk Information Group
INFO_GROUP_FLASH	NVRAM Flash Information Group
INFO_GROUP_TEMPERATURE	NVRAM Temperature Sensors Information Group
INFO_GROUP_DDR	NVRAM DDR Information Group
INFO_GROUP_BACKUP_POWER	NVRAM Backup Power Information Group
INFO_GROUP_PCIE	NVRAM PCIe Information Group
INFO_GROUP_DEBUG	NVRAM Debug Information Group

3.3.2 Card Status Get

Prototype

```
status_get <group>
```

Description

This command gets the dynamic status for the NVRAM card by group.

Parameters

Parameter	Type	Description	Values
group	Mandatory	ID of the status group	For general information, see the following table. For detailed descriptions, see the section "NVRAM Status Group" in the *NV1600 Flashtec™ NVRAM Drive Programmer's Manual* [2].

Output

Status values for each group.

Table 3 • NVRAM Status Group Values

Field	Description
STATUS_GROUP_GENERAL	General Status Group
STATUS_GROUP_CPU	CPU Status Group
STATUS_GROUP_SYSTEM	System Status Group
STATUS_GROUP_DEBUG	Debug Status Group **Note:** Currently, the group contains no data. Calling status_get with this value will return “unsupported value”.
STATUS_GROUP_FLASH	Flash Status Group
STATUS_GROUP_BACKUP_POWER	Backup Power Status Group
STATUS_GROUP_PCIE	PCIe Status Group

3.3.3 Master Authenticate

Prototype

```
master_authenticate <auth_key>
```

Description

If authentication has been enabled as described in the section "Authentication" of the *NV1600 Flashtec NVRAM Drives Programmer's Manual* [2] , this command runs master authentication between the Host and NVRAM card. With the exception of admin_authenticate and a few others, most CLI commands cannot be issued until master authentication is performed.

Note:
If authentication is not enabled, all CLI commands are available.

Parameters

Parameter	Type	Description	Values
auth_key	Mandatory	Master authentication key	32 characters long ASCII string

3.3.4 Admin Authenticate

Prototype

```
admin_authenticate <auth_key>
```

Description

If authentication has been enabled as described in the section "Authentication" of the *NV1600 Flashtec NVRAM Drives Programmer's Manual* [2] , this command runs admin authentication between the Host and NVRAM card. Typically, this command is used to recover the master authentication key.

Parameters

Parameter	Type	Description	Values
auth_key	Mandatory	Admin authentication key	32 characters long ASCII string

3.3.5 Configuration Set

Prototype

```
set_config <config_type> <config data>
```

Description

This command sets the NVRAM card configuration.

Parameters

Parameter	Type	Description	Values
config_type	Mandatory	ID of configuration field.	See the following table.
config_data	Mandatory	Configuration data.	See the following table.

Writable Configuration Info Groups

Field	Description	Data type
ACTIVE_FW_INDEX	The active firmware index	Integer (0 – 3)
ENABLE_BIST	Enable/Disable BIST in the next init (Built-In Self-Test)	String (TRUE or FALSE)
AUTHENTICATION_KEY_MASTER	Master authentication key.	String of 32 characters
AUTHENTICATION_KEY_ADMIN	Admin authentication key.	String of 32 characters
NVRAM_CONFIG_TYPE_CPU_TEMPERATURE_THRESHOLD	CPU temperature threshold that triggers a "temperature" event when crossed	Integer (Kelvin degrees)
NVRAM_CONFIG_TYPE_CPU_CRITICAL_TEMPERATURE_THRESHOLD	CPU critical temperature threshold that triggers a "temperature" event when crossed	Integer (Kelvin degrees)
NVRAM_CONFIG_TYPE_SYSTEM_TEMPERATURE_THRESHOLD	System temperature threshold that triggers a "temperature" event when crossed	Integer (Kelvin degrees)
NVRAM_CONFIG_TYPE_SYSTEM_CRITICAL_TEMPERATURE_THRESHOLD	System critical temperature threshold that triggers a "temperature" event when crossed	Integer (Kelvin degrees)
NVRAM_CONFIG_TYPE_BACKUP_POWER_TEMPERATURE_THRESHOLD	Backup Power temperature threshold that triggers a "temperature" alarm when crossed	Integer (Kelvin degrees)
AUTO_RESTORE_BANK	Default bank to load restore data **Note:** The API returns P_STATUS_NVME_FLASH_OPERATED error if a flash operation, such as Erase, Backup, Restore and Bad Block Scan, already running.	String (BANK_0 or BANK_1)
AUTO_RESTORE_MODE	NVRAM auto restore mode **Note:** The API returns P_STATUS_NVME_FLASH_OPERATED error if a flash operation, such as Erase, Backup, Restore and Bad Block Scan, already running.	String (AUTO_RESTORE_MODE_DISABLE or AUTO_RESTORE_FROM_BACKUP or AUTO_RESTORE_FROM_BANK)
HEARTBEAT_MSG_INTERVAL	Heartbeat message interval	Integer (1-10 seconds)
HEARTBEAT_MSG_NUMBER	Heartbeat number of lost heartbeats that indicate a necessity of vaulting	Integer (3-7 messages)

Field	Description	Data type
AUTO_BACKUP_MODE_WHEN_POWER_LOST	Enable/Disable NVRAM auto backup **Note:** The API returns P_STATUS_NVME_FLASH_OPERATED error if a flash operation, such as Erase, Backup, Restore and Bad Block Scan, already running.	String (AUTO_BACKUP_MODE_DISABLE or AUTO_BACKUP_MODE_ENSABLE)
AUTO_BACKUP_BANK_WHEN_POWER_LOST	ID of the flash bank that will be used to store the RAM backup on the next power down **Note:** The API returns P_STATUS_NVME_FLASH_OPERATED error if a flash operation, such as Erase, Backup, Restore and Bad Block Scan, already running.	String (BANK_0 or BANK_1)
AUTO_BACKUP_MODE_WHEN_HEARTBEAT_LOST	NVRAM heartbeat loss backup mode **Note:** The API returns P_STATUS_NVME_FLASH_OPERATED error if a flash operation, such as Erase, Backup, Restore and Bad Block Scan, already running.	String (AUTO_BACKUP_MODE_DISABLE or AUTO_BACKUP_MODE_ENABLE)
AUTO_BACKUP_BANK_WHEN_HEARTBEAT_LOST	ID of the flash bank that will be used to store the RAM backup on the heartbeat loss **Note:** The API returns P_STATUS_NVME_FLASH_OPERATED error if a flash operation, such as Erase, Backup, Restore and Bad Block Scan, already running.	String (BANK_0 or BANK_1)
AUTO_BACKUP_MODE_WHEN_CPU_OVER_TEMPERATURE	Enable/Disable auto backup when CPU over temperature **Note:** The API returns P_STATUS_NVME_FLASH_OPERATED error if a flash operation, such as Erase, Backup, Restore and Bad Block Scan, already running.	String (AUTO_BACKUP_MODE_DISABLE or AUTO_BACKUP_MODE_ENSABLE)
AUTO_BACKUP_BANK_WHEN_CPU_OVER_TEMPERATURE	Auto backup bank ID when CPU over temperature **Note:** The API returns P_STATUS_NVME_FLASH_OPERATED error if a flash operation, such as Erase, Backup, Restore and Bad Block Scan, already running.	String (BANK_0 or BANK_1)
AUTO_BACKUP_MODE_WHEN_SYSTEM_OVER_TEMPERATURE	Enable/Disable auto backup when System over temperature **Note:** The API returns P_STATUS_NVME_FLASH_OPERATED error if a flash operation, such as Erase, Backup, Restore and Bad Block Scan, already running.	String (AUTO_BACKUP_MODE_DISABLE or AUTO_BACKUP_MODE_ENSABLE)

Field	Description	Data type
AUTO_BACKUP_BANK_WHEN_SYSTEM_OVER_TEMPERATURE	Auto backup bank ID when System over temperature **Note:** The API returns P_STATUS_NVME_FLASH_OPERATED error if a flash operation, such as Erase, Backup, Restore and Bad Block Scan, already running.	String (BANK_0 or BANK_1)
AUTO_BACKUP_MODE_WHEN_BACKUP_POWER_OVER_TEMPERATURE	Enable/Disable auto backup when Backup Power over temperature **Note:** The API returns P_STATUS_NVME_FLASH_OPERATED error if a flash operation, such as Erase, Backup, Restore and Bad Block Scan, already running.	String (AUTO_BACKUP_MODE_DISABLE or AUTO_BACKUP_MODE_ENSABLE)
AUTO_BACKUP_BANK_WHEN_BACKUP_POWER_OVER_TEMPERATURE	Auto backup bank ID when Backup Power over temperature **Note:** The API returns P_STATUS_NVME_FLASH_OPERATED error if a flash operation, such as Erase, Backup, Restore and Bad Block Scan, already running.	String (BANK_0 or BANK_1)
ENABLE_RAMDISK_ENCRYPTION	Enable/Disable RAM Disk encryption. The change will take effect after next restart. **Note:** Encryption can only be enabled after setting the Encryption key. See Set the RAM Disk Encryption Key on page 30.	String (TRUE or FALSE)
ENABLE_RAMDISK_DMI_OVERLAP	Enable/Disable RAM Disk and DMI overlap. The change will take effect after next restart.	String (TRUE or FALSE)
DMI_SIZE	The size of the DMI when overlap is disabled. The change will take effect after next restart. See section "NVRAM Config Data" in the *NV1600 Flashtec NVRAM Drives Programmer's Manual* [2] for a detailed description of the constraints with respect to the DMI size value.	Integer (1-0x400000000 bytes)
RAMDISK_SIZE	The size of the RAM Disk when overlap is disabled. The change will take effect after next restart. See section "NVRAM Config Data" in the *NV1600 Flashtec NVRAM Drives Programmer's Manual* [2] for a detailed description of the constraints with respect to the RAM Disk size value.	Integer (1-0x400000000 bytes)
BAD_BLOCK_SCAN_USE_VBBS	Enable/Disable the use of the VBBS table in bad block scan	String (TRUE or FALSE)

Field	Description	Data type
ENABLE_DEBUG	Enable/Disable debug functions. **Note:** Can only be set when Admin is authenticated.	String (TRUE or FALSE)

3.3.6 Configuration Get

Prototype

```
get_config <config_type>
```

Description

This command gets the NVRAM card configuration.

Parameters

Parameter	Type	Description	Values
config_type	Mandatory	ID of configuration type	See Configuration Set on page 17.

3.3.7 Scan Flash Bad Blocks

Prototype

```
bad_block_scan  <bank_id>
```

Description

This command starts a bad block scanning operation on a flash bank. When finished, the event is received if you had registered for the event (CLI wait_for_event command).

Parameters

Parameter	Type	Description	Values
bank_id	Mandatory	ID of bank to scan	String (BANK_0 or BANK_1).

3.3.8 Erase Flash Bank

Prototype

```
erase  <bank_id> <erase_type>
```

Description

This command starts erasing a flash bank. When finished, the event is received if you had registered for the event (CLI wait_for_event command).

Parameters

Parameter	Type	Description	Values
bank_id	Mandatory	ID of bank to erase	String (BANK_0 or BANK_1).
erase_type	Mandatory	Type of erase: standard or secure (erase, fill with 0 and erase again).	String (NVRAM_FLASH_ERASE_TYPE_STANDARD or NVRAM_FLASH_ERASE_TYPE_SECURE).

3.3.9 Backup

Prototype

```
backup  <bank_id>
```

Description

This command starts backup of RAM contents to a flash bank. When finished, the event is received if you had registered for the event (CLI wait_for_event command).

Parameters

Parameter	Type	Description	Values
bank_id	Mandatory	ID of bank to backup	String (BANK_0 or BANK_1).

3.3.10 Get Backup Results

Prototype

```
backup_results_get  <bank_id>
```

Description

This command retrieves information about the last backup operation that was run.

Parameters

Parameter	Type	Description	Values
bank_id	Mandatory	ID of bank	String (BANK_0 or BANK_1).

Output

Backup information of the chosen flash bank.

1. Backup ID of the last backup
2. Total number of DDR UC errors that were detected during last backup
3. List of the first up to 32 DDR UC errors (offset, size) that were detected during last backup

3.3.11 Restore

Prototype

```
restore  <bank_id>
```

Description

This command starts restoring of the RAM content from a flash bank. When finished, the event is received.

Parameters

Parameter	Type	Description	Values
bank_id	Mandatory	ID of bank to restore	String (BANK_0 or BANK_1).

3.3.12 Flash Bank Info

Prototype

```
bank_info  <bank_id>
```

Description

This command retrieves information about flash bank.

Parameters

Parameter	Type	Description	Values
bank_id	Mandatory	ID of bank	String (BANK_0 or BANK_1).

Output

Information of chosen flash bank.

3.3.13 VBB (Vendor Bad Block) Count Get

Prototype

```
vbb_count_get <bank_id>
```

Description

This command returns the number of vendor bad blocks found in a flash bank.

Parameters

Parameter	Type	Description	Values
bank_id	Mandatory	ID of bank to return number of vendor bad blocks.	String (BANK_0 or BANK_1)

3.3.14 Restore Results Get

Prototype

```
restore_results_get
```

Description

The command returns information about the last restore operation that was run.

Parameters

None.

Output

1. Timestamp of the last restore operation
2. Number of DDR chunks that were not restored during last restore
3. The restored size (in bytes) from the last restore operation
4. The backup image size (in bytes)
5. List of the DDR chunks (offset, size) that were not restored.

3.3.15 Register Event Handler

Prototype

```
reg_ev_handler  <event> <handler> <operation>
```

Description

This command registers/unregisters the default handler for an event type. The CLI handles the default event handler for each type of event.

Parameters

Parameter	Type	Description	Values
event	Mandatory	Event type	See below. For details, see "NVRAM Event Type" in the *NV1600 Flashtec™ NVRAM Drive Programmer's Manual* [2].
handler	Mandatory	Handler type	Each handler type is represented by the enumeration listed in the table below.

Parameter	Type	Description	Values
			Note: Microsemi recommends using the same type of handler as the the type of event.
operation	Mandatory	Register or unregister	String (TRUE or FALSE)

Event Type Values

Event type	Handler type	Description
EVENT_TYPE_VAULT	NVRAM_EV_HANDLER_VAULT	Event sent when a vault operation has finished (backup, restore, erase, bad block scan).
EVENT_TYPE_CPU	NVRAM_EV_HANDLER_CPU	Event sent when a CPU event occurs.
EVENT_TYPE_SYSTEM	NVRAM_EV_HANDLER_SYSTEM	Event sent when a system event occurs.
EVENT_TYPE_BACKUP_POWER	NVRAM_EV_HANDLER_BACKUP_POWER	Event sent when a backup power component event occurs.
EVENT_TYPE_DDR_ECC	NVRAM_EV_HANDLER_DDR_ECC	Event sent when a DDR uncorrectable ECC error occurs.
EVENT_TYPE_GENERAL	NVRAM_EV_HANDLER_GENERAL	Event sent when a general event occurs.
NVRAM_EVENT_TYPE_UC_ERROR	NVRAM_EV_HANDLER_UC_ERROR	Event sent when an uncorrectable error event occurs.

3.3.16 Time of Day Sync

Prototype

```
time_of_day_sync
```

Description

This command synchronizes the time of day for the card to the current host time.

Parameters

None

3.3.17 Card Statistics Get

Prototype

```
statistics_get <group>
```

Description

This command gets statistics from the NVRAM card according to group.

Parameters

Parameter	Type	Description	Values
group	Mandatory	ID of statistics group	For the group values, see the following table. For a detailed description, see the section "NVRAM Statistics Group" in the *NV1600 Flashtec™ NVRAM Drive Programmer's Manual* [2].

Output

Statistic values for each group.

NVRAM Statistics Group Values

Field	Description
STATISTICS_GROUP_FLASH	NAND flash statistics group.
STATISTICS_GROUP_DDR	DDR statistics group.
STATISTICS_GROUP_GENERAL	General statistics group.

3.3.18 Card Statistics Reset

Prototype

```
statistics_reset <group>
```

Description

This command resets statistics for the NVRAM card. The list of groups is described in the *NV1600 Flashtec™ NVRAM Drive Programmer's Manual*.

Parameters

Parameter	Type	Description	Values
group	Mandatory	ID of statistics group	See Card Statistics Get on page 25. For detailed description please refer to *NV1600 Flashtec™ NVRAM Drive Programmer's Manual* [2].

3.3.19 Download SBL

Prototype

```
sbl_download  <slot> <type> <path>
```

Description

This command downloads the SBL (secondary boot loader) to the card.

Parameters

Parameter	Type	Description	Values
slot	Ignored	Slot to download	String (FW_SLOT_1 or FW_SLOT_2 or FW_SLOT_3) **Note:** Although the value is ignored and will not be processed, this field must include a legal value for proper operation.
type	Mandatory	Download source type	String (DOWNLOAD_FROM_FILE or DOWNLOAD_FROM_MEM)
path	Mandatory	Path of SBL EEPROM Image to download	String

3.3.20 Download Firmware Image

Prototype

```
firmware_download  <slot> <type> <path>
```

Description

This command downloads the firmware file to the card.

Parameters

Parameter	Type	Description	Values
slot	Mandatory	Slot to download	String (FW_SLOT_2 or FW_SLOT_3)
type	Mandatory	Download source type	String (DOWNLOAD_FROM_FILE or DOWNLOAD_FROM_MEM)
path	Mandatory	Path to firmware image file	String

3.3.21 Firmware Metadata Get

Prototype

```
fw_metadata_get
```

Description

This command retrieves firmware metadata information.

Parameters

None.

3.3.22 Heartbeat Management

Prototype

```
heart_beat <command>
```

Description

This command starts/stops heartbeat transmission from the host to the card.

Parameters

Parameter	Type	Description	Values
command	Mandatory	Start or stop the heart beat sending	String (see below)

NVRAM Heartbeat Command Values

Field	Description
HEARTBEAT_COMMAND_START	Starts heartbeat transmission.
HEARTBEAT_COMMAND_STOP	Stop heartbeat transmission from the host and listening to it in the card.
HEARTBEAT_COMMAND_ABORT	Aborts heartbeat transmission.

3.3.23 Reboot the Card

Prototype

```
reboot
```

Description

This command reboots the card.

Parameters

None

3.3.24 Read PFI Data

Prototype

```
pf_info
```

Description

This command reads and displays the Product Feature Info data.

Example:

```
PMCNVM/NVRAM/CARD 0 >pf_info
HW REV: Test_Ver_RevD
Encryption: 1
Data: 100001
Board REV: RevD
```

Note:
If PFI data was not programmed to EEPROM, an error code is returned. See the *NV1600 Flashtec NVRAM Drives Programmer's Manual* [2] for details about error codes.

Parameters

None

3.3.25 Get the LBA to DDR Address

Prototype

```
lba_to_addr_map_get <lba>
```

Description

This command gets the LBA to DDR address mapping (that is, gets the DDR address of a specific LBA number) when RAMDISK/DMI overlap is enabled.

Parameters

Parameter	Type	Description	Values
lba	Mandatory	The LBA number	Integer.

3.3.26 Get the DDR Address to LBA

Prototype

```
addr_to_lba_map_get <addr>
```

Description

This command gets DDR address to LBA mapping (that is, gets the LBA number of a specific DDR Address) when RAMDISK/DMI overlap is enabled.

Parameters

Parameter	Type	Description	Values
addr	Mandatory	The DDR Address	Integer (64 bit).

3.3.27 Set the RAM Disk Encryption Key

Prototype

```
ramdisk_enc_key <key1> <key2>
```

Description

This command sets the RAM Disk encryption keys. They keys are write only and cannot be retrieved.

Example:

```
ramdisk_enc_key 11112222333344445555666677778888 00010002000300040005000600070008
```

Parameters

Parameter	Type	Description	Values
key1	Mandatory	First key **Note:** Each 32-bits (4 bytes) of the key MUST be unique and non-zero.	Integer
key2	Mandatory	Second key **Note:** Each 32-bits (4 bytes) of the key MUST be unique and non-zero.	Integer

3.3.28 Restore Corrupted

Prototype

```
restore_corrupted  <bank_id>
```

Description

This command starts restoring the image of the RAM content from a corrupted flash bank.

Parameters

Parameter	Type	Description	Values
bank_id	Mandatory	ID of bank to restore	String (BANK_0 or BANK_1).

3.3.29 Self Test

Prototype

```
self_test
```

Description

This command starts a self test of the card.

Parameters

None

3.3.30 Enter the Memory Level

Prototype

```
mem
```

Description

Changes to the Memory commands directory level.

Parameters

None

3.3.31 Enter the RAM Disk Level

Prototype

```
ramdisk
```

Description

Changes to the Ram Disk commands directory level.

Parameters

None

3.3.32 Enter the Debug Level

Prototype

```
debug
```

Description

Changes to the Debug commands directory level.

Note: This level is “hidden”. It is not displayed by “help” command.

Parameters

None

3.4 Memory-Level Commands

The memory level displays the prompt, "PMCNVM/NVRAM/CARD #/MEM >".

3.4.1 Memory Map

Prototype

```
memmap <size> <permissions>
```

Description

This command maps the NVRAM memory into host virtual memory.

Other memory commands are allowed only after issuing this command. The list of permissions is described in the *NV1600 Flashtec™ NVRAM Drive Programmer's Manual* [2].

Parameters

Parameter	Type	Description	Values
size	Mandatory	Size of mapped memory in MB	Integer(1024 – 15360)
permissions	Mandatory	Memory access permissions	String ("READ" or "WRITE" or "READ|WRITE" or "EXEC" or "EXEC|READ" or "EXEC|WRITE" or "EXEC|READ|WRITE)

3.4.2 Write to Memory

Prototype

```
write_mem <offset>  <size> <data>
```

Description

This command writes the pattern to the mapped DMI. Memory writes are performed in 8-byte chunks. They start from the offset (size/8) times the given value.

Parameters

Parameter	Type	Description	Values
offset	Mandatory	Memory offset in bytes	Integer
size	Mandatory	Size of memory to write in bytes	Integer
data	Mandatory	Data to write	Integer

3.4.3 Read from Memory

Prototype

```
read_mem <offset>  <size> [t]
```

Description

Reads the mapped DMI. Use the "t" option to read the memory as text.

Parameters

Parameter	Type	Description	Values
offset	Mandatory	Memory offset in bytes	Integer
size	Mandatory	Size of memory to read from in bytes	Integer
t	Optional	Reads the memory as text	't' – read the memory as text

3.4.4 Reset Whole Memory

Prototype

```
reset_whole_mem
```

Description

This command resets whole memory of the card. The mapped DMI is filled with the pattern "0x535455565758595A".

Parameters

None

3.4.5 Set Memory From File

Prototype

```
set_mem_from_file <file>  <interval> <times>
```

Description

This command sets the location of the sample data file, the size of the DMI memory interval segments, and specified number of times to repeat the sequence.

Parameters

Parameter	Type	Description	Values
file	Mandatory	Path to the file containing the sample data.	String
interval	Mandatory	The size of the DMI memory interval segments between samples (in bytes).	Integer
times	Mandatory	Number of times to repeat the sequence.	Integer

3.4.6 Dump Memory To File

Prototype

```
dump_mem_to_file <file>  <offset> <size>
```

Description

This command reads from card memory to file.

Parameters

Parameter	Type	Description	Values
file	Mandatory	Path to file to read to	String
offset	Mandatory	Memory offset (in bytes)	Integer
size	Mandatory	Size of memory to read from (in bytes)	Integer

3.4.7 Compare Memory to File

Prototype

```
compare_mem_to_file <file>  <start> <size> <times>
```

Description

This command compares card memory to the contents of a file.

Parameters

Parameter	Type	Description	Values
file	Mandatory	Path to file to compare to	String
start	Mandatory	Start of memory area to compare (in bytes)	Integer
size	Mandatory	Size of memory area to compare (in bytes)	Integer
times	Mandatory	Number of times to repeat the sequence	Integer

3.5 RAM Disk-Level Commands

The RAM disk level displays the prompt, "PMCNVM/NVRAM/CARD #/RAMDISK >".

3.5.1 Format RAM Disk

Prototype

```
format
```

Description

This command formats the RAM disk with the ext4 Linux filesystem.

Parameters

None

3.5.2 Mount the RAM Disk

Prototype

```
mount  <path>
```

Description

This command mounts the RAM disk to the path.

Parameters

Parameter	Type	Description	Values
path	Mandatory	Path to mount the RAM disk	String

3.5.3 Unmount the RAM Disk

Prototype

```
unmount  <path>
```

Description

This command unmounts the RAM disk from the path.

Parameters

Parameter	Type	Description	Values
path	Mandatory	Path to mount the ram disk	String

3.5.4 Copy File to RAM Disk

Prototype

```
copy  <path> <times> [<destination>]
```

Description

This command copies a file to the RAM disk a predefined number of times.

Parameters

Parameter	Type	Description	Values
path	Mandatory	Path to file to copy	String
times	Mandatory	Number of times to copy the file	Integer
destination	Optional	Destination path in the RAM disk. If not set, the file is copied to the RAM disk root	String

3.5.5 Delete File from RAM Disk

Prototype

```
delete  <path>
```

Description

This command deletes the file from the RAM disk.

Parameters

Parameter	Type	Description	Values
path	Mandatory	Path to file to delete	String

3.5.6 List RAM Disk Files

Prototype

```
list  [path]
```

Description

This command lists the files of the RAM disk.

Parameters

Parameter	Type	Description	Values
path	Optional	Path to list	String

3.6 Debug-Level Commands

The debug level displays the prompt, "PMCNVM/NVRAM/CARD #/DEBUG >".

Note:

1. By default, the debug commands are disabled. To enable the debug commands, authenticate the card first; see "Setting the Master and Admin Authentication Keys" in the *NV1600 Flashtec NVRAM Drives Installation and User's Guide* [1].
2. By default, the debug level is “hidden” and is not displayed by the “help” command.

3.6.1 Print Log

Prototype

```
print_log <log_page_id> <num_strings>
```

Description

This command prints logs by page ID. The page ID is a log page for a specific node or the ECC errors statistics history page. See the *NV1600 Flashtec™ NVRAM Drive Programmer's Manual* [2] for more details about nodes.

Parameters

Parameter	Type	Description	Values
log_page_id	Mandatory	The ID of the node log page	String ("LOG_PAGE_PROC_34" or "LOG_PAGE_PROC_12" or "LOG_PAGE_PROC_14" or "LOG_PAGE_PROC_15" or "LOG_PAGE_PROC_21" or "LOG_PAGE_PROC_22" or "LOG_PAGE_PROC_24" or "LOG_PAGE_PROC_25" or "LOG_PAGE_PROC_33" or "LOG_PAGE_PROC_11" or "LOG_PAGE_PROC_35" or "LOG_PAGE_PROC_36" or "LOG_PAGE_PROC_43" or "LOG_PAGE_PROC_44" or "LOG_PAGE_PROC_45" or "LOG_PAGE_PROC_46" or "LOG_PAGE_STAT_HISTORY”)
num_strings	Mandatory	Number of strings	Integer (1 – 960)

3.6.2 Print Backup Log

Prototype

```
print_backup_log <log_page_id> <num_strings>
```

Description

This command prints backup logs by page ID. See the *NV1600 Flashtec™ NVRAM Drive Programmer's Manual* [2] for more details about nodes.

Parameters

Parameter	Type	Description	Values
log_page_id	Mandatory	The ID if Node log page	String (see Print Log on page 37)
num_strings	Mandatory	Number of strings	Integer (1 – 960)

3.6.3 Log Polling

Prototype

```
log_polling  <operation >
```

Description

This command enables/disables output of all log pages.

Parameters

Parameter	Type	Description	Values
operation	Mandatory	Start or stop	String (TRUE or FALSE)

3.6.4 Read DDR

Prototype

```
read_ddr <address> <size>
```

Description

This command reads from card DDR memory. The <address> parameter must be a real address, known by the firmware.

Parameters

Parameter	Type	Description	Values
address	Mandatory	Real card DDR address in bytes	Integer

Parameter	Type	Description	Values
size	Mandatory	Size of DDR to read from in bytes	Integer

3.6.5 Get FRIM Mapping

Prototype

```
get_frim_mapping  <bank_id> <chs> <targs> <luns>
```

Description

This command prints the bad blocks map per LUN / all.

Parameters

Parameter	Type	Description	Values
bank_id	Mandatory	ID of bank to get mapping	String (BANK_0 or BANK_1).
chs	Mandatory	channels	Integer (0xff for all)
targs	Mandatory	targets	Integer (0xff for all)
luns	Mandatory	LUNs	Integer (0xff for all)

3.6.6 Clear FRIM Mapping

Prototype

```
clear_frim_mapping  <bank_id>
```

Description

This command clears the FRIM block mapping table.

Parameters

Parameter	Type	Description	Values
bank_id	Mandatory	ID of bank to clear frim	String (BANK_0 or BANK_1).

3.6.7 Register Access

Prototype

```
register  <set / get / ddr_set> <node  | address> <value> <mask>
```

Description

This command reads/writes to the CSR registers.

Parameters

Parameter	Type	Description	Values
set/get/ ddr_set	Mandatory	Type of access.	Integer (0—register get, 1—register set, 2—ddr set)
address	Mandatory	Absolute address of the register or absolute address in DDR.	Integer for register access (superposition of Node << 24 | address). Integer64 for DDR access.
value	Optional only for register set and ddr set	Value to write.	Integer
mask	Optional only for register set	Bits to read, modify, write.	Integer

3.6.8 Dump DDR to File

Prototype

```
dump_ddr_to_file <address> <size> <file>
```

Description

This command copies DDR content to a file. The <address> parameter must be a real address, known by the firmware.

Parameters

Parameter	Type	Description	Values
address	Mandatory	Real card DDR offset in bytes	Integer
size	Mandatory	Size of DDR to dump in bytes	Integer
file	Mandatory	Path to file to write to	String

3.6.9 Test Firmware Fail

Prototype

```
fw_fail <proc> <op>
```

Description

This command is a debug command for testing.

Parameters

Parameter	Type	Description	Values
proc	Mandatory	ID of processor to fail	String (see Print Log on page 37).
Op	Mandatory	Fail operation	String (see below)

Fail Operation Values

Field	Description
FW_FAIL_TEST_DIV0	Simulate firmware division by 0
FW_FAIL_TEST_HANG	Simulate firmware hanging
FW_FAIL_TEST_DIV0_NEXT_INIT	Simulate firmware division by 0 during the next re-initialization of the firmware
FW_FAIL_TEST_HANG_NEXT_INIT	Simulate firmware hanging during the next re-initialization of the firmware

3.6.10 Dump Flash Block to File

Prototype

```
dump_block_to_file < channel> <target> <lun> <block> <file>
```

Description

This command dumps the specified flash block contents to a file.

Parameters

Parameter	Type	Description	Values
channel	Mandatory	Channel of the block to dump	Integer
target	Mandatory	Target of the block to dump	Integer
lun	Mandatory	LUN of the block to dump	Integer
block	Mandatory	Block to dump	Integer
file	Mandatory	Path to file to dump to	String

3.6.11 Dump Flash Page to File

Prototype

```
dump_page_to_file < channel> <target> <lun> <block> <page> <file>
```

Description

This command dumps the specified flash page contents to a file.

Parameters

Parameter	Type	Description	Values
channel	Mandatory	Channel of the page to dump	Integer
target	Mandatory	Target of the page to dump	Integer
lun	Mandatory	LUN of the page to dump	Integer
block	Mandatory	Block of the page to dump	Integer
page	Mandatory	Page to dump	Integer
file	Mandatory	Path to file to dump to	String

3.6.12 Inject Uncorrectable Errors

Prototype

```
inject_UC_errors <bank_id> <start_channel> <start_target> <start_lun>
<start_block> <start_page> <skip_channel> <skip_target> <skip_lun> <skip_block>
 <skip_page>
```

Description

This command injects uncorrectable errors while reading from flash.

Parameters

Parameter	Type	Description	Values
bank_id	Mandatory	ID of flash bank to inject	String (BANK_0 or BANK_1).
start_channel	Mandatory	Start channel to inject error	Integer
start_target	Mandatory	Start target to inject error	Integer
start_lun	Mandatory	Start LUN to inject error	Integer
start_block	Mandatory	Start block to inject error	Integer
start_page	Mandatory	Start page to inject error	Integer
skip_channel	Mandatory	Channel to skip	Integer
skip_target	Mandatory	Target to skip	Integer
skip_lun	Mandatory	LUN to skip	Integer
skip_block	Mandatory	Block to skip	Integer
skip_page	Mandatory	Page to skip	Integer

3.6.13 Set the Number of Bad Blocks

Prototype

```
bad_block_count_set  <bad_block_count> <value>
```

Description

This command sets the number of flash blocks as “bad” or “unmapped”.

Parameters

Parameter	Type	Description	Values
bad_block_count	Mandatory	Number of blocks to set	Integer
value	Mandatory	Value to set	Integer(1 – bad, 0 – unmapped)

3.6.14 Set the Number of Bad Blocks by LUN

Prototype

```
bad_block_set_by_lun  <bank> <num_of_blocks> <channel> <target> <lun>
<first_block> <set_as_bad>
```

Description

This command sets the number of blocks in a specific LUN as “bad” or “unmapped”.

Parameters

Parameter	Type	Description	Values
bank	Mandatory	ID of flash bank to set blocks	String (BANK_0 or BANK_1).
num_of_blocks	Mandatory	Number of blocks to set	Integer
channel	Mandatory	Start channel to set	Integer
target	Mandatory	Start target to set	Integer
lun	Mandatory	Start LUN to set	Integer
first_block	Mandatory	Start block to set	Integer
set_as_bad	Mandatory	Value to set	Integer(1 – bad, 0 – unmapped)

3.6.15 Write to Flash Block from File

Prototype

```
write_to_block_from_file  <channel> <target> <lun> <block> <file>
```

Description

This command reads a block content from a file and writes it to a specific flash block.

Note: The source file size must be exactly as a DDR block size 16640*256= **4259840**.

Parameters

Parameter	Type	Description	Values
channel	Mandatory	Channel of the block to write	Integer
target	Mandatory	Target of the block to write	Integer
lun	Mandatory	LUN of the block to write	Integer
block	Mandatory	Block to write	Integer
file	Mandatory	Path to file to read	String

3.6.16 Debug Configuration Set

Prototype

```
debug_config_set <config_type> <config_data>
```

Description

This command sets the NVRAM debug configuration.

Parameters

Parameter	Type	Description	Values
config_type	Mandatory	The configuration type to set	See below.
config_data	Mandatory	The configuration data to set	See below.

NVRAM Debug Configuration Type

Field	Description	Range
NVRAM_DEBUG_CONFIG_TYPE_IGNORE_BAD_BLOCK_THRESHOLD	Ignore bad block limit threshold. **Note:** The change will take effect after next restart.	true or false

3.6.17 Debug Configuration Get

Prototype

```
debug_config_get <config_type>
```

Description

This command gets the NVRAM debug configuration.

Parameters

Parameter	Type	Description	Values
config_type	Mandatory	The configuration type to get	See below.

NVRAM Debug Configuration Type

Field	Description	Range
NVRAM_DEBUG_CONFIG_TYPE_IGNORE_BAD_BLOCK_THRESHOLD	Reads if the bad block limit threshold is ignored.	true or false

3.6.18 Wait For Event

Prototype

```
wait_for_event [abort_only]
```

Description

The command returns information about the last restore operation that was run.

Parameters

Parameter	Type	Description	Values
abort_only	Optional	Abort Only Flag.	TRUE: Do NOT start a new "waiting", only stop a previous "waiting" if it exists

Note: This call is only used for debug purposes in the CLI. The wait_for_event is included in the event registration process.

A Usage Example

The following is an example for how to use the CLI:

1. From a Linux shell, launch the CLI program:

```
$ su - root
# <release_root>/demo/pmcnvm
```

2. At the in-program prompt, specify the NVRAM level:

```
PMCNVM >nvram
```

3. List the installed cards:

```
PMCNVM/NVRAM >card_list
0 Card UID: 0400001084AE
```

4. Initialize Card 0:

```
PMCNVM/NVRAM >init 0
Card 0 init finished successfully
```

5. Enter the Card 0 level:

```
PMCNVM/NVRAM >card 0
```

6. Enter the Memory level:

```
PMCNVM/NVRAM/CARD 0 >mem
```

7. Perform the desired tasks.

a MICROCHIP company

Microsemi Headquarters
One Enterprise, Aliso Viejo,
CA 92656 USA

Within the USA: +1 (800) 713-4113
Outside the USA: +1 (949) 380-6100
Fax: +1 (949) 215-4996
Email: sales.support@microsemi.com
www.microsemi.com

Microsemi, a wholly owned subsidiary of Microchip Technology Inc. (Nasdaq: MCHP), offers a comprehensive portfolio of semiconductor and system solutions for aerospace & defense, communications, data center and industrial markets. Products include high-performance and radiation-hardened analog mixed-signal integrated circuits, FPGAs, SoCs and ASICs; power management products; timing and synchronization devices and precise time solutions, setting the world's standard for time; voice processing devices; RF solutions; discrete components; enterprise storage and communication solutions; security technologies and scalable anti-tamper products; Ethernet solutions; Power-over-Ethernet ICs and midspans; as well as custom design capabilities and services. Learn more at **www.microsemi.com**.

PMC-2152852

The technology discussed in this document may be protected by one or more patent grants.