Mellanox Firmware Tools (MFT)

User's Manual

Rev 0.40

Mellanox Firmware Tools User's Manual

Document Number: 2329

Mellanox Technologies, Inc.
2900 Stender Way
Santa Clara, CA 95054
U.S.A.
www.Mellanox.com

Tel: (408) 970-3400
Fax: (408) 970-3403

Mellanox Technologies Ltd
PO Box 586 Hermon Building
Yokneam 20692
Israel

Tel: +972-4-909-7200
Fax: +972-4-959-3245

Table of Contents

Revision History

Table 1 - Revision History Table

Date	Revision	Description
January 2007	0.40	• MFT for Windows is now part of the WinIB software package; therefore, to install MFT on a Windows machine, you need to install WinIB and enable MFT. See Section 1.5.2, "Install MFT On Windows OS," on page 9. • Added ***flint*** flag: -use_image_ps • Removed ***flint*** flags: -crc, -bsn
January 2006	0.30	(MFT version 1.0.1) • Added querying options for VPD for mlxburn • Added examples to demonstrate support of MT43132 InfiniScale device by mlxburn and spark • Reorganized the "flint - HCA Firmware Burner' (page 15) chapter • Added the Appendix "PSID Assignment" on page 23.
October 2005	0.20	Added Windows distribution to MFT (MFT version 0.5.1)
		Added the following sections: • Section 1.1, "Supported Operating Systems," on page 7 • Section 1.3, "MFT Access to Hardware Devices," on page 8 • Section 1.4, "Software Prerequisites," on page 9 • Section 1.5, "MFT Installation," on page 9
August 2005	0.10	First release (Linux distribution only) (MFT version 0.5.0)

1 Introduction

The Mellanox Firmware Tools (MFT) package is a set of firmware management tools for a single InfiniBand node. MFT can be used for:

- Generating a standard or customized Mellanox firmware image
- Querying for firmware information
- Burning a firmware image to a single InfiniBand node

The following is a list of the available tools in MFT, together with a brief description of what each tool performs.

mlxburn This tool provides the following functions:
- Generation of a standard or customized Mellanox firmware image for burning (in binary or .mlx format)
- Burning an image to the Flash/EEPROM attached to a Mellanox HCA or switch device
- Querying the firmware version loaded on an HCA board
- Displaying the VPD (Vital Product Data) of an HCA board

flint This tool burns a firmware *binary* image to the Flash(es) attached to an HCA board. It includes query functions to the burnt firmware image and to the binary image file.

spark This tool burns a firmware *binary* image to the EEPROM(s) attached to a switch device. It includes query functions to the burnt firmware image and to the binary image file.

1.1 Supported Operating Systems

Two distributions of MFT are available: for Linux and for Windows. Please refer to the release notes of your version for supported platforms and kernels.

Note: Unless explicitly specified, the usage of the tools is identical for both operating systems.

1.2 Supported Mellanox Devices

Table 1 - Supported Mellanox Devices

Device Type	Product Name
HCA	MT23108 InfiniHost
	MT25208 InfiniHost III Ex
	MT25204 InfiniHost III Lx
Switch	MT47396 InfiniScale III
	MT43132 InfiniScale

1.3 MFT Access to Hardware Devices

The MFT tools access a hardware device using its assigned *mst device name*. These names get assigned in the following manner:

On Linux:
run ‘mst start’

On Windows:
1. To access an HCA device via the PCI bus: If the IBAL driver is installed and running, the names are already assigned.
2. To access an HCA or switch device via the USB bus: A USB to I2C Adapter should be used to connect the host USB port and the I2C port of the target device. Upon the first usage of this interface, you will be requested to install the USB to I2C Adapter driver (I2CBridge[1]).

To list the available mst device names, run 'mst status' (on both OSes).

mst device name format:

`mt<dev-id>_pci<_crX|confX>`

where:

X is the index of the HCA on the machine.
_crX devices access the HCA directly (recommended if possible)
confX devices use configuration cycles to access the HCA

1. Visit http://www.xdimax.com to download this driver.

1.4 Software Prerequisites

1.4.1 On Linux OS

Table 2 - MFT Software Dependencies on Linux

Software Package	Required Version
Perl	5.6 or later
Expat	1.95 or later
zlib	1.1.4 or later

1.4.2 On Windows

Table 3 - MFT Software Dependencies on Windows

Software Package	Required Version
WinIB	1.2.0 or later
I2CBridge[1] (Dimax's Driver for USB to I2C Adapter)	0.1.4 or later

1. Visit http://www.xdimax.com to download this driver.

1.5 MFT Installation

1.5.1 Install MFT On Linux OS

1. Download the Linux MFT package from the Mellanox Firmware webpage (http://www.mellanox.com/support/firmware_download.php).
2. Untar the downloaded package
3. Run 'install.sh'
4. Start the mst driver by entering: mst start

   Note: It is possible to customize some parameters of the installation (such as the target installation path). Run 'install.sh --help' for details.

1.5.1.1 Uninstall MFT (Linux)

To uninstall MFT, run 'uninstall_mft.sh' (located under the same directory of the other MFT executables).

1.5.2 Install MFT On Windows OS

MFT for Windows is provided as part of the Mellanox Technologies WinIB software package. If WinIB is installed on your machine[1] and you wish to check whether MFT is installed too, open a console window and run: '`mlxburn -v`'.

In case an error is returned by the command, then MFT is not installed on your machine. To install it, activate the 'Add or Remove Programs' utility of Windows and click the 'Change' button. Select 'Modify' on the Program Maintenance pop-up menu, and enable the following components:

- Driver
- MFT

1. That is, you can locate the program 'MellanoxWinIB32/64' using the 'Add or Remove Programs' utility of Windows.

In case WinIB is not installed on your machine, download it via http://www.mellanox.com and perform either a ‘Custom’ or a ‘Complete’ installation. If you choose a Complete installation of WinIB, MFT will be installed in the process. If you choose a Custom installation, make sure you enable the Driver and MFT components.

1.5.2.1 Uninstall MFT (Windows)

To *uinstall* MFT, activate the ‘Add or Remove Programs’ utility of Windows and click the ‘Change’ button. Select ‘Modify’ on the Program Maintenance pop-up menu, and disable MFT.

2 mlxburn - FW Image Generator & Burner

2.1 Overview

mlxburn is a tool for firmware (FW) image generation and/or for burning a firmware image to the Flash/EEPROM attached to a Mellanox device. Both functions or a single function of **mlxburn** can be activated by means of command line options (see Section 2.2, "mlxburn Synopsys"). It can also query the FW version and VPD info of HCA cards.

mlxburn allows for cusomization of standard Mellanox firmware for OEM specific needs (e.g., a specific HCA board type). The customization parameters are specified using .ini files a unique parameter-set ID (PSID).

2.2 mlxburn Synopsys

mlxburn [-h][-V][-v] <-dev mst-device | -wrimage fw-image-file> <-fw mellanox-fw-file | -image fw-image-file [-format BINARY|IMG]> [-conf parameter-set-file][-nofs][-nofs_img][-dev_type device-number][-exp_rom <exp_rom_file>][-fwver] [-force] [-conf_dir <dir>] [-vpd] [-vpd_rw]

where:

-h - displays a short help text

-V <INFORM|WARNING|DEBUG>- sets the verbosity level. Default is WARNING.

-v - prints version info and exits

-dev <mst-dev> - burns the image onto the device with the provided MST device name

-fw <mellanox-fw-file> - specifies the Mellanox firmware file to use (file extension is .mlx)

-image <fw-image-file> - uses the given firmware image to burn (file extension is .bin or .img)

-conf <parameter-set-file> - firmware configuration file (.ini). Needed for image generation (not using the -dev flag) or if auto-detection of configuration fails

-wrimage <fw-image-file> - writes the image to the provided file name.

-nofs - when specified, the burn process will not be failsafe. A non-failsafe burn is required (on the rare occasion) when a new firmware version has modifications in the Invariant Sector

-nofs_img - when specified, the generated image will not be failsafe. If burning is also specified, it will not be failsafe either.

-format <BINARY|IMAGE> - specifies which image format to use. Can be used only with the -wrimage flag. Default is BINARY.

-dev_type <Mellanox-Device-ID>- **mlxburn** must know the Mellanox device ID in order to work properly. This option should be used if auto-detection of the device type (taken from the firmware file) fails. The following is the list of supported device IDs:

23108 - For MT23108 InfiniHost based HCA cards (Cougar family)

25208 - For MT25208 InfiniHost III Ex in InfiniHost-mode (with local attached memory) HCA cards (Lion Cub family)

25218 - For MT25208 InfiniHost III Ex in MemFree-mode HCA cards (Lion Mini family)

25204 - For MT25204 InfiniHost III Lx HCA cards (Tiger/Cheetah families)

43132 - For MT43132 InfiniScale based switch systems

47396 - For MT47396 InfiniScale III based switch systems

-exp_rom <exp_rom_file>	- integrates the given expansion rom file into the firmware image. The given file may be in .img or .bin (raw binary) format. Note that the exp_rom_en (expansion ROM enable FW) parameter must be set in the configuration file.
-fwver	- returns the loaded firmware version on the Mellanox device.
-force	- runs mlxburn in non-interactive mode
-conf_dir <dir>	- instructs the burn operation to look for auto-detected configuration files under the specified directory rather than under the firmware file directory.
-vpd	- displays the Read Only section of the PCI VPD (Vital Product Data) of the given device. See Notes 1, 2 below.
	Note: The VPD feature may not be supported on certain board types.
-vpd_rw	- displays the Read Only and Read/Write sections of the PCI VPD of the given device. See Notes 1, 3 below.
	Note: The VPD feature may not be supported on certain board types.

Notes:
1. The VPD feature may not be supported on certain board types.
2. (Linux Only): This option requires running as root.
3. (Windows Only): This option is currently not supported for Windows.

2.3 Tool Description

The **mlxburn** firmware update flow is composed of two separate stages: image generation and image burning. In the image generation stage a given Mellanox firmware release (in .mlx) is processed to generate a 'burnable' firmware image. This image is burnt to a the Flash/EERPROM attached to a Mellanox device in the second stage. The burning process retains device specific data such as GUIDs, VSD, and BSN. Also, the burn process is failsafe by default.

mlxburn runs both stages by default, but it may perform only one of the stages by means of command options. If the '-dev' option is not specified (see Section 2.2, "mlxburn Synopsys"), the actual device burning is skipped. Specifying the '-image' option skips the image generation stage and loads the provided image (generated in a previous run of **mlxburn** using the '-wrimage' option).

Note: When generating an image file for a Mellanox switch device, the produced image file name must end with a .img extension.

2.3.1 Firmware Customization

A Mellanox firmware release can be customized (usually) to fit a specific board type. The customization is done by using a FW parameter-set file in the image generation stage. This file has a .ini format. Each parameter-set file has a unique parameter-set ID (PSID), which is kept in the device Flash/EEPROM and allows retaining device configuration during future FW updates.

During a device FW update, **mlxburn** reads the PSID from the device and uses the corresponding .ini file when generating the FW image. **mlxburn** searches for the files in the same directory of the FW release. When **mlxburn** is used to generate an image file, or when no corresponding parameter-set file is found, the user should explicitly specify which parameter-set file to use.

To produce an image file the user needs to provide the option '-wrimage <target file>'. To actually burn the image to the Flash/EEPROM attached to a Mellanox HCA or switch device, the user needs to specify the option '-dev </dev/mst/dev-file>'.

If run in burning mode, **mlxburn** auto-detects the firmware parameter-set with which the device was previously burnt. It locates and uses this parameter-set file to generate the appropriate image for the device (by merging the FW release with the specific parameter-set required).

To inhibit image generation, the '-image <pre-generated-image-file>' should be used. It instructs **mlxburn** to use the given file for burning the device.

2.4 Examples

- To update firmware on an MT23108 InfiniHost (HCA) device with the configuration file (.ini) auto-detected, enter:
```
mlxburn -fw ./fw-23108-a1-rel.mlx -dev /dev/mst/mt23108_pci_cr0
```

- To generate a failsafe image file for the same HCA above without burning, enter:
```
mlxburn -fw ./fw-23108-a1-rel.mlx -conf ./MHX-CE128-T.ini -wrimage ./fw-23108.bin
```

- To update firmware on an MT23108 InfiniHost (HCA) device with the configuration file (.ini) explicitly specified, enter:
```
mlxburn -fw ./fw-23108-a1-rel.mlx -conf ./MHX-CE128-T.ini
      -dev /dev/mst/mt23108_pci_cr0
```

- To burn a firmware binary file for an MHGA28-1T (Lion Cub 128MB DDR) HCA card, enter:
```
mlxburn -image ./fw-25208-4_7_400-MHGA28-1T.bin -dev /dev/mst/mt25208_pci_cr0
      -dev_type 25208
```

- To update firmware on an MT47396 InfiniScale III device, enter:
```
mlxburn -fw IS3FW.BIN -dev /dev/mst/mtusb-1
```
  Note: This firmware update cannot be performed before initializing the MST device (mtusb-1) to connect to the I2C-compatible bus of the InfiniScale III and its EEPROM.

- To generate an image for the InfiniScale III switch device, enter:
```
mlxburn -fw IS3FW.BIN -conf ./MTS2400-A00.INI -wrimage IS3FW.img
```
  Note: The generated firmware image to be burnt to a switch device must have a '.img' file name extension.

- To update firmware on an MT43132 InfiniScale device in a switch system such as Flextronics' F-X430066 Stallion 8 4X IB port switch, enter:
```
mlxburn -image Stallion_5_5_0.eeprom -dev /dev/mst/mtusb-1 -dev_type 43132
```
  Note: This firmware update cannot be performed before initializing the MST device (mtusb-1) to connect to the I2C-compatible bus of the InfiniScale and its EEPROM.

2.5 Exit Return Values

The following exit values are returned:

0 - successful completion

>0 - an error occurred

3 flint - HCA Firmware Burner

3.1 Overview

The **flint** (Flash interface) utility performs the following functions:

- Burns a binary firmware image to the Flash attached to an HCA device
- Queries for HCA firmware attributes (version, GUIDs, PSID, etc.)
- Enables executing various operations on the Flash memory from the command line (for debug/production)

3.2 flint Synopsys

flint [switches...] <command> [parameters...]

3.2.1 Switch Descriptions

-d[evice] <device> - The device to which the Flash is connected.

Affected commands: All

-i[mage] <image> - Binary image file.

Affected commands: burn, verify

-guid <GUID> - Base value for up to 4 GUIDs which are automatically assigned the following values:

guid -> node GUID
guid+1 -> port1
guid+2 -> port2
guid+3 -> system image GUID.

Affected commands: burn

-guids <GUIDs...> - 4 GUIDs must be specified here. These GUIDs will be assigned to:

node, port1, port2 and system image GUID respectively.

Affected commands: burn

-clear_semaphore - Force the clearing of the Flash semaphore on the device. This flag should come BEFORE the -d[evice] flag in the command line. No command is allowed when this flag is used.

NOTE: Using this flag may result in an unstable behavior and flash image corruption if the device or another flash application is currently using the flash. Handle with care.

-byte_mode - Shift the address when accessing Flash internal registers. May be required for burn/write commands when accessing certain Flash types.

-h[elp] - Print this message and exit.

-hh - Print extended command help.

-nofs - Do not burn image in failsafe mode.

-skip_is	- Allow burning the FW image without updating the invariant sector to ensure fail-safe burning even if the invariant sector of the image is different from the one burnt on the Flash. Note: Some FW releases may include important changes to the invariant sector that must be included in the FW update process. In these cases, the -skip_is flag should *not* be used. Please refer to the specific FW release notes for details.
-s[ilent]	- Print errors only. Affected commands: burn
-y[es]	- Non-interactive mode. Assume the answer "yes" to all questions. Affected commands: all
-vsd <vendor-specific-data>	-A VSD string, composed of up to 208 characters, will be written to the VSD section in the flash. If not specified, the current VSD will be preserved.
-use_image_ps	- Burn vsd as it appears in the given image. (Default: Retain current vsd on Flash.) Affected commands: burn
-v	- Version information.

3.2.2 Command Descriptions

The **flint** utility commands are:

b[urn] - Burn flash

e[rase] - Erase sector

q[uery] - Query misc. flash/FW characteristics

rw - Read one dword from flash

v[erify] - Verify entire flash

ww - Write one dword to flash

bb - Burn Block - Burns the given image as is. No checks are done.

wwne - Write one dword to flash without sector erase

wbne - Write a data block to flash without sector erase

rb - Read a data block from flash

ri - Read the fw image on the flash.

dc - Dump Configuration: print fw configuration file for the given image.

The following sections provide the command line syntax for the following **flint** utility commands, together with examples of usage.

- Burning FW (page 17)
- Querying the FW Image (page 17)
- Verifying the FW Image (page 18)

3.2.2.1 Burning FW

The FLINT utility enables you to burn the Flash from a binary image.

To burn the entire Flash from a raw binary image, use the following command line:

```
flint -d <device> -i <fw-file> [-guid <GUID> | -guids <4 GUIDS>] burn
```

where:

device – Device on which the flash is burned.

fw-file – Binary firmware file.

GUID (optional) – One or four GUIDs.

If 4 GUIDS are provided (-guids flag), they will be assigned as node, port1, port2 and system image GUIDs, respectively.
If only one GUID is provided (-guid flag), it will be assigned as node GUID. Its values +1, +2 and +3 will be assigned as port1, port2 and system image GUID, respectively.
If no -guid/-guids flag is provided, the currect GUIDs will be preserved on the device.

Note: A GUID is a 16-digit hexadecimal number. If less than 16 digits are provided, leading zeros will be inserted.

Examples:

1. Update the FW on the device, keeping the current GUIDs and VSD. (Note: This is the common way to use flint.)
   ```
   > flint -d /dev/mst/mt23108_pci_cr0 -i fw-23108-3_3_5-MHXL-CF128-T.bin burn
   ```

2. Update the FW on the device, specifying the GUIDs to burn.
   ```
   > flint -d /dev/mst/mt23108_pci_cr0 -i fw-23108-3_3_5-MHXL-CF128-T.bin -guid
   12345678deadbeef burn
   ```

3. Burn the image on a blank Flash device. This means that no GUIDs are cuurently burnt on the device, therefore they must be supplied (with -guid/-guids) by the burning command. Moreover, the burn process cannot be fail-safe when burning a blank Flash, therefore the -nofs flag must be specified.
   ```
   > flint -d /dev/mst/mt23108_pci_cr0 -i fw-23108-3_3_5-MHXL-CF128-T.bin -nofs -guid
   1234567812345678 burn
   ```

4. Read FW from the device and save it as an image file.
   ```
   > flint -d /dev/mst/mt23108_pci_cr0 ri Flash_Image_Copy.bin
   ```

3.2.2.2 Querying the FW Image

To query FW image on a device, use the following command line:

```
flint -d <device> q
```

To query FW image in a file, use the following command line:

```
flint -i <image file> q
```

where:

device – Device on which the query is run.

image file – Image file on which the query is run.

Examples:

1. Query the FW on the device.
   ```
   > flint -d /dev/mst/mt23108_pci_cr0 query
   ```

2. Query the FW image file.
   ```
   > flint -i fw-23108-3_3_5-MHXL-CF128-T.bin query
   ```

3.2.2.3 Verifying the FW Image

To verify the FW image on the Flash, use the following command line:

```
flint -d <device> v
```

To verify the FW image in a file, use the following command line:

```
flint -i <image file> v
```

where:

device – the Flash device to verify

image file – the image file to verify

Examples:

```
flint -d /dev/mst/mt23108_pci_cr v
flint -i ./image_file.bin v
```

3.2.3 Additional Debug / Development Commands

3.2.3.1 Reading a Word from Flash

To read one dword from Flash memory, use the following command line:

```
flint -d <device> rw addr
```

where:

device – the device the dword is read from.

addr – the address of the word to read.

Example:

```
flint -d /dev/mst/mt23108_pci_cr0 rw 0x20
```

3.2.3.2 Writing a dword to Flash

To write one dword to Flash memory, use the following command line:

```
flint -d <device> ww addr data
```

where:

device – the device the dword is written to.

addr – the address of the word to write.

data – the value of the word.

Example:

```
flint -d /dev/mst/mt23108_pci_conf01 ww 0x10008 0x5a445a44
```

3.2.3.3 Writing a dword to Flash Without Sector Erase

To write one dword to Flash memory without sector erase , use the following command line:

```
flint -d <device> wwne addr data
```

where:

device – the device the dword is written to.

addr – the address of the word to write.

data – the value of the word.

Example:

```
flint -d /dev/mst/mt23108_pci_cr0 wwne 0x10008 0x5a445a44
```

Note that a result may be dependent on the Flash type. Usually, bitwise and between the specified word and the previous Flash contents will be written to the specified address.

3.2.3.4 Erasing a Sector

To erase a sector that contains a specified address, use the following command line:

```
flint -d <device> e addr
```

where:

device – the device the sector is erased from.

addr – the address of a word in the sector that you want to erase.

Example:

```
flint -d /dev/mst/mtusb-1 e 0x1000
```

4 spark - Switch Firmware Burner

4.1 Overview

The **spark** tool enables burning a binary firmware image to the EEPROM residing on an InfiniScale III (MT47396) board.

4.2 spark Synopsys

spark [switches...] <command> [parameters...]

where switches are:

-d[evice] <device>	- defines the Mellanox device to which the EEPROM is connected. Affected commands: All (see the commands below)
-i[mage] <image>	- Image file is in ".img" format. Affected commands: burn, verify, query
-guid <GUID>	- Uses the given guid as the node guid of the burnt image. By default, the guid is taken from the image on the EEPROM. Affected commands: burn
-sysguid <GUID>	- Use the given guid as the system image guid of the burnt image. By default, this value is taken from the current image on the EEPROM. Affected commands: burn
-bsn <BSN>	- Mellanox Board Serial Number (BSN). The vallid BSN format is: MTxxxxx[-]R[xx]ddmmyy-nnn[-cc]. By default, this value is taken from the current image on the EEPROM. Affected commands: burn
-ndesc <Descr>	- Use the given string (max 64 characters) as the node description. By default, this value is taken from the current image on the EEPROM. Affected commands: burn
-is3_i2c <i2c_addr>	- Provides the I2C address of the switch device. If this flag is not specified, then the the default address for Mellanox switch devices is: 0x6c.
-pe_i2c <i2c_addr>	- Provides the I2C address of the primary EEPROM. By default, this address is read from the Mellanox switch device. Use this flag only if the switch device is not accessible.
-se_i2c <i2c_addr>	- Provides the I2C address of the secondary EEPROM. By default, this address is read from the Mellanox switch device. Use this flag only if the switch device is not accessible.
-h[elp]	- Prints this help message and exits.
-hh	- Prints an extended command help
-nofs	- Do *not* burn the firmware image in failsafe mode.
-s[ilent]	- Print errors only.

Affected commands: burn

-y[es] - Non-interactive mode. Assume the answer to all questions is "yes".

Affected commands: All

-v - Version information.

The commands of **spark** are:

b[urn] - Burns the binary image to the EEPROM.

Parameters: None

Examples:

spark -d /dev/mst/mtusb-1 -i image1.img burn

spark -d /dev/mst/mtusb-1 -guid 0x2c9000100d050 -i image1.img b

q[uery] - Queries miscellaneous EEPROM and firmware characteristics

Parameters: None

Example:

spark -d /dev/mst/mtusb-1 query

v[erify] - Verifies the entire EEPROM

Parameters: None

Example:

spark -d /dev/mst/mtusb-1 v

bb - Burns the given image as is (Burn Block). No checks are performed on EEPROM or on the given image. Also no fields (e.g., BSN or GUIDs) are read from the EEPROM.

Parameters: None

Example:

spark -d /dev/mst/mtusb-1 -i image1.img bb

ri - Reads the firmware image on the EEPROM and writes it to a file.

Parameters: filename to write the image to (in .img format)

Example:

spark -d /dev/mst/mtusb-1 ri file.img

rb - Reads a block of data from a single eeprom to the given file.

Parameters: <eeprom address> <start offset> <data size> <output file name>

Example:

spark -d /dev/mst/mtusb-1 rb 0x56 0x0 0x1000 out.img

Appendix A. PSID Assignment

In some cases, OEMs or board manufacturers may wish to use a specific FW configuration not supplied by Mellanox. After setting the new FW parameters in an INI file, the user should assign a unique PSID (Parameter Set ID) to this new configuration. The PSID is kept as part of the FW image on the device NVMEM. The firmware burning tools use this field to retain FW settings while updating FW versions.

This appendix explains how to assign a new PSID for a user customized FW, and how to indicate to the burning tools that a new PSID exists.

> Note: Please change FW parameters with caution. A faulty setting of FW parameters may result in undefined behavior of the burnt device.

A.1 PSID Field Structure

The PSID field is a 16-ascii (byte) character string. If the assigned PSID length is less then 16 characters, the remaining characters are filled with binary 0s by the burning tool.

Table 1 provides the format of a PSID.

Table 1 - PSID format

Vendor symbol	Board Type Symbol	Board Version Symbol	Parameter Set Number	Reserved
3 characters	3 characters	3 characters	4 characters	3 characters (filled with '\0')

Example: A PSID for Mellanox's MHXL-CF128-T HCA board is MT_0030000001, where:

MT_	Mellanox vendor symbol
003	MHXL-CF128-T board symbol
000	Board version symbol
0001	Parameter Set Number

A.2 PSID Assignment and Integration Flow

To assign and integrate the new PSID to produce the new FW

1. Write the new FW configuration file (in .INI format).
2. Assign it with a PSID in the format described above. Use your own vendor symbol to assure PSID uniqueness. If you do not know your vendor symbol, please contact your local Mellanox FAE.
3. Set the PSID parameter in the new FW configuration file.