



**Release Notes**

# MT23108 InfiniHost & MT25208 InfiniHost III Ex HCA Device Driver for Linux

Rev 3.2.0-IBGD

Mellanox Technologies

© Copyright 2004. Mellanox Technologies, Inc. All Rights Reserved.

MT23108 InfiniHost and MT25208 InfiniHost III Ex HCA Device Driver Release Notes For Linux

**Document Number:**

Mellanox Technologies, Inc.  
2900 Stender Way  
Santa Clara, CA 95054  
U.S.A.  
[www.Mellanox.com](http://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

Mellanox Technologies

# 1 Overview

This is the Rev 3.2.0-IBGD release of the InfiniHost (MT23108) and InfiniHost III Ex (MT25208) HCA driver. It is very similar to Rev 3.2.0.

## 1.1 Main Changes from version 3.2.0

- Integration with Linux kernel Makefiles.
- API change: Added two new entries to the structure `VAPI_hca_vendor_t` that is used in `VAPI_query_hca_cap` function:  

```
u_int8_t pci_bus; /* PCI bus HCA is on */  
u_int8_t pci_devfn; /* PCI device/function of HCA */
```
- All code of non-Linux operating systems (Darwin, Windows, etc.) was cleaned.

## 1.2 Supported Platforms and Operating Systems

For a list of supported platforms and operating systems, please refer to the *Mellanox IB Gold Distribution Release Notes*.

## 1.3 Supported InfiniHost Firmware Versions

This release has been tested with:

- InfiniHost firmware version fw-23108-rel-3.2.0
- InfiniHost III Ex firmware version fw-25208-rel-4.5.3

## 1.4 Supported InfiniHost Based Hardware

This release was tested with the following HCA boards:

- MHX-CEXXX-T<sup>1</sup> (previously MTPB23108) InfiniHost PCI-X HCA Adapter Card
- MHXL-CFXXX-T<sup>1</sup> (previously MTLP23108) Low Profile InfiniHost PCI-X HCA Adapter Card
- MHEL-CFXXX-T<sup>1</sup> (previously MTLP25208) InfiniHost III Ex HCA Adapter Card

## 1.5 Verbs Supported

For the definition of the verbs, see the document: *Mellanox IB-Verbs API (VAPI), Rev. 1.0, Doc. #AN010601062*.

See `vapi.h` and `evapi.h` for all verbs supported.

---

1. XXX reflects the size of on-board memory (in MB): 128, 256, or 512.

## 2 Limitations and Known Bugs

In addition to bugs and limitations documented in the *InfiniHost MT23108-A1 Errata rev 1.50, Doc. #2060GI*, the next two tables list the bugs and limitations of the VAPI driver.

Table 1 - Recent Limitations and Known Bugs

	Description	Details
1.	An application that uses fork and is multi-threaded must perform the fork only in the main thread.	Performing the fork in any other thread can cause the application to hang.
2.	An application that uses fork must register the data buffers in full pages.	Otherwise, the application can incur a segmentation fault.
3.	A kernel consumer that needs to register IO-memory must use physical registration only.	There is no way to identify IO memory for kernel addresses. Virtual registration of IO-memory is supported for user applications.
4.	If an application attaches and registers a shared memory region but detaches from it without de-registering it in the HCA, then it is possible that this memory region will get assigned to another application as shared memory. In such a state, it is possible that the HCA will perform now-invalid write operations to this same region. This state holds until the driver's cleanup code closes all the resources.	The reason this occurs is that the linux driver for shared-memory big pages sets values to the page descriptors of the shared memory physical pages. As a result, it is not possible to use <code>get_page()/put_page()</code> to protect the physical pages until de-registration of the memory region is called.
5.	User level application may lose event notifications in case of slow event processing by the process.	Each user space process has a limited buffer for queuing event notifications. If the process stalls too long in an event callback, this buffer may become full and then any additional event destined to this process will be dropped. The event queue has been enlarged to 512, but the scenario may still occur.
6.	QP WQE buffer may use non full pages. In case of fork usage, this can cause segmentation fault in the child process.	To be fixed in next release.
7.	Huge pages are not supported on SuSE SLES 9.0 operating system	To be fixed in rev. 4.0.0

## **3 Fixed Bugs**

Description	Details
The number of kernel UD AVs was limited to 256	The limit is now 2048

Mellanox Technologies

Mellanox Technologies