

Cnuas Verbs Provider, Datasheet



rdma-core user space verbs provider for the CnuasNIC adapter

Document DS-CNU-014 • **Revision** B • **Issued** 23 August 2026 • **Status** Published

Item	Value
Part	libcnuas-rdmav34.so
Type	rdma-core user space provider
Provider name	cnuas
Package	cnuas-libcnuas-provider
Version	v0.1.0-27-gc7e8629
Repo	PacketFive/CnuasNIC

1. Overview

The Cnuas verbs provider is the user space half of the RDMA stack. It plugs into the standard rdma-core library so that libibverbs applications, perftest, rdma_cm users and NCCL reach a CnuasNIC through the same calls they use for physical adapters. No application changes and no LD_PRELOAD shims are involved: rdma-core discovers the provider from a driver description file at startup and matches it to the cnuas_ib kernel device.

Key features

- Discovered by stock rdma-core, no patched libibverbs required.
- Complete queue pair, completion queue and memory region life cycle.
- Shared receive queue and multicast attachment.
- Address handle creation for datagram traffic.
- Versioned kernel to user ABI, refused on mismatch.
- Standard `ibv_reg_dmabuf_mr()` reaches the CnuasNIC kernel DMA-BUF MR operation without a vendor-specific verbs command.

2. Discovery and loading

Item	Value
Driver description file	cnuas.driver, containing driver cnuas
Installed location	The rdma-core provider directory, typically <code>/etc/libibverbs.d/</code>
Shared object	libcnuas-rdmav34.so
Registration macro	<code>PROVIDER_DRIVER(cnuas, cnuas_dev_ops)</code>
Kernel ABI version	1, accepted as both minimum and maximum
Kernel counterpart	cnuas_ib.ko, see Kernel Modules

The `rdmav34` suffix follows the rdma-core provider naming convention and records the provider ABI generation the library is built for.



ABI mismatch is a refusal, not a fault

The provider declares the same value as its minimum and maximum accepted kernel ABI. If the loaded `cnuas_ib.ko` reports a different version the provider declines the device and `ibv_devinfo` shows no adapter, rather than binding and failing later in an unclear way. Keep the module and the provider from the same release.

3. Verbs implemented

Group	Operations
Device and port	<code>query_device_ex</code> , <code>query_port</code>
Protection domain	<code>alloc_pd</code> , <code>dealloc_pd</code>
Memory region	<code>reg_mr</code> , standard <code>ibv_reg_dmabuf_mr</code> , <code>dereg_mr</code>
Completion queue	<code>create_cq</code> , <code>destroy_cq</code> , <code>poll_cq</code> , <code>req_notify_cq</code>
Queue pair	<code>create_qp</code> , <code>modify_qp</code> , <code>destroy_qp</code>
Data path	<code>post_send</code> , <code>post_recv</code>
Shared receive queue	<code>create_srq</code> , <code>modify_srq</code> , <code>query_srq</code> , <code>destroy_srq</code> , <code>post_srq_recv</code>
Address handle	<code>create_ah</code> , <code>destroy_ah</code>
Multicast	<code>attach_mcast</code> , <code>detach_mcast</code>

Device limits that applications read through `query_device_ex` come from the kernel module and are tabulated in Kernel Modules section 4.1.

4. Application compatibility

Consumer	Status
<code>ibv_devinfo</code> , <code>ibv_devices</code>	Works
<code>ib_send_bw</code> , <code>ib_write_bw</code> , <code>ib_read_bw</code> , <code>ib_send_lat</code>	Works
rdma and ip tooling	Works
NCCL over the network transport	Works, see <code>tests/test_nccl_link.py</code>
<code>opensm</code>	Not supported, no user space management datagram device exists
Vendor specific verbs extensions	Not supported

5. Validation

Feature	Test
RoCEv2 end to end, two guests through the switch	<code>tests/test_rocev2_e2e.py</code>
Native InfiniBand end to end	<code>tests/test_ib_e2e.py</code>
Collective transport over the fabric	<code>tests/test_nccl_link.py</code>

These are integration tests: a meaningful run needs the switch daemon, two guest virtual machines and the provider at once, which is why they live in the superproject rather than in `PacketFive/CnuasNIC`. Current results are recorded in the Validation Matrix.



6. Integration information

Item	Value
Repo	PacketFive/CnuasNIC
Submodule path	src/cnuasnic
Source	src/cnuasnic/userspace/, src/cnuasnic/rdma-core/
Build	Make for the standalone object, CMake through rdma_provider() when built inside an rdma-core tree
Debian package	cnuas-libcnuas-provider
Depends on	libibverbs, cnuas_ib.ko
Language	C

7. Revision history

Revision	Notes
A	First publication. Operation list read from the provider operations table in the source.
B	Recorded the upstream DMA-BUF verbs registration path used by libcnuaspeerem.

PacketFive, Packet Five Networks Ltd., Dublin, Ireland.

Cnuas Virtual AI/HPC Infrastructure is published at <https://github.com/PacketFive/cnuas> under the Apache License 2.0; read it at <https://github.com/PacketFive/cnuas/blob/main/LICENSE>.

Cnuas Virtual AI/HPC Infrastructure is emulation software. It is not affiliated with, endorsed by, or derived from any hardware vendor, and every device it models is a software artefact. The work is published by its authors in a personal capacity and is not sponsored or endorsed by any employer.

Specifications describe the referenced revision of the software and may change without notice. Contact info@packetfive.com.