

**NAME**

`ibv_open_device`, `ibv_close_device` – open and close an RDMA device context

**SYNOPSIS**

```
#include <infiniband/verbs.h>
```

```
struct ibv_context *ibv_open_device(struct ibv_device *device);
```

```
int ibv_close_device(struct ibv_context *context);
```

**DESCRIPTION**

`ibv_open_device()` opens the device *device* and creates a context for further use.

`ibv_close_device()` closes the device context *context*.

**RETURN VALUE**

`ibv_open_device()` returns a pointer to the allocated device context, or NULL if the request fails.

`ibv_close_device()` returns 0 on success, -1 on failure.

**NOTES**

`ibv_close_device()` does not release all the resources allocated using context *context*. To avoid resource leaks, the user should release all associated resources before closing a context.

Setting the environment variable `**RDMAV_ALLOW_DISASSOC_DESTROY**` tells the library to relate an EIO from destroy commands as a success as the kernel resources were already released. This comes to prevent memory leakage in the user space area upon device disassociation. Applications using this flag cannot call `ibv_get_cq_event` or `ibv_get_async_event` concurrently with any call to an object destruction function.

**SEE ALSO**

`ibv_get_device_list(3)`, `ibv_query_device(3)`, `ibv_query_port(3)`, `ibv_query_gid(3)`, `ibv_query_pkey(3)`

**AUTHORS**

Dotan Barak <dotanba@gmail.com>