NAME

ibv_alloc_td(), ibv_dealloc_td() − allocate and deallocate thread domain object

SYNOPSIS

#include <infiniband/verbs.h>

struct ibv_td *ibv_alloc_td(struct ibv_context **context*,
struct ibv_td_init_attr **init_attr*);

int ibv_dealloc_td(struct ibv_td **td*);

DESCRIPTION

ibv_alloc_td() allocates a thread domain object for the RDMA device context *context*.

The thread domain object defines how the verbs libraries and provider will use locks and additional hardware capabilities to achieve best performance for handling multi-thread or single-thread protection. An application assigns verbs resources to a thread domain when it creates a verbs object.

If the *ibv_td* object is specified then any objects created under this thread domain will disable internal locking designed to protect against concurrent access to that object from multiple user threads. By default all verbs objects are safe for multi-threaded access, whether or not a thread domain is specified.

A *struct ibv_td* can be added to a parent domain via **ibv_alloc_parent_domain()** and then the parent domain can be used to create verbs objects.

ibv_dealloc_td() will deallocate the thread domain *td*. All resources created with the *td* should be destroyed prior to deallocating the *td*.

RETURN VALUE

ibv_alloc_td() returns a pointer to the allocated struct *ibv_td* object, or NULL if the request fails (and sets errno to indicate the failure reason).

ibv_dealloc_td() returns 0 on success, or the value of errno on failure (which indicates the failure reason).

SEE ALSO

ibv_alloc_parent_domain(3),

AUTHORS

Alex Rosenbaum <alexr@mellanox.com>

Yishai Hadas <yishaih@mellanox.com>