NAME

ibv_alloc_mw, ibv_dealloc_mw − allocate or deallocate a memory window (MW)

SYNOPSIS

#include <infiniband/verbs.h>

struct ibv_mw *ibv_alloc_mw(struct ibv_pd **pd*,
enum ibv_mw_type *type*);

int ibv_dealloc_mw(struct ibv_mw **mw*);

DESCRIPTION

ibv_alloc_mw() allocates a memory window (MW) associated with the protection domain *pd*. The MW's type (1 or 2A/2B) is *type*.

The MW is created not bound. For it to be useful, the MW must be bound, through either ibv_bind_mw (type 1) or a special WR (type 2). Once bound, the memory window allows RDMA (remote) access to a subset of the MR to which it was bound, until invalidated by: ibv_bind_mw verb with zero length for type 1, IBV_WR_LOCAL_INV/IBV_WR_SEND_WITH_INV WR opcode for type 2, deallocation.

ibv_dealloc_mw() Unbinds in case was previously bound and deallocates the MW *mw*.

RETURN VALUE

ibv_alloc_mw() returns a pointer to the allocated MW, or NULL if the request fails. The remote key (**R_Key**) field **rkey** is used by remote processes to perform Atomic and RDMA operations. This key will be changed during bind operations. The remote process places this **rkey** as the rkey field of struct ibv_send_wr passed to the ibv_post_send function.

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

NOTES

ibv_dereg_mr() fails if any memory window is still bound to this MR.

SEE ALSO

ibv_alloc_pd(3), **ibv_post_send**(3), **ibv_bind_mw**(3), **ibv_reg_mr**(3),

AUTHORS

Majd Dibbiny <majd@mellanox.com>

Yishai Hadas <yishaih@mellanox.com>