## NAME

ibv_alloc_parent_domain(), ibv_dealloc_pd() − allocate and deallocate the parent domain object

## SYNOPSIS

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

**struct ibv_pd *ibv_alloc_parent_domain(struct ibv_context** **context",* **, struct ibv_parent_domain_init_attr** **attr*);

## DESCRIPTION

**ibv_alloc_parent_domain()** allocates a parent domain object for the RDMA device context *context.*

The parent domain object extends the normal protection domain with additional objects, such as a thread domain.

A parent domain is completely interchangeable with the *struct ibv_pd* used to create it, and can be used as an input argument to any function accepting a *struct ibv_pd.*

The behavior of each verb may be different if the verb is passed a parent domain *struct ibv_pd* that contains a *struct ibv_td pointer.* For instance the verb my choose to share resources between objects using the same thread domain. The exact behavior is provider dependent.

The *attr* argument specifies the following:

```
struct ibv_parent_domain_init_attr {
                        struct ibv_pd *pd; /* referance to a protection domain, can't be NULL */
                        struct ibv_td *td; /* referance to a thread domain, or NULL */
                        uint32_t comp_mask;
};
```

**ibv_dealloc_pd()** will deallocate the parent domain as its exposed as an ibv_pd *pd.* All resources created with the parent domain should be destroyed prior to deallocating the parent domain.

## RETURN VALUE

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

## SEE ALSO

**ibv_alloc_parent_domain**(3), **ibv_dealloc_pd**(3), **ibv_alloc_pd**(3), **ibv_alloc_td**(3)

## AUTHORS

Alex Rosenbaum <alexr@mellanox.com>

Yishai Hadas <yishaih@mellanox.com>