

NAME

`ibv_create_qp`, `ibv_destroy_qp` – create or destroy a queue pair (QP)

SYNOPSIS

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

```
struct ibv_qp *ibv_create_qp(struct ibv_pd *pd,
                             struct ibv_qp_init_attr *qp_init_attr);
```

```
int ibv_destroy_qp(struct ibv_qp *qp);
```

DESCRIPTION

ibv_create_qp() creates a queue pair (QP) associated with the protection domain *pd*. The argument *qp_init_attr* is an `ibv_qp_init_attr` struct, as defined in `<infiniband/verbs.h>`.

```
struct ibv_qp_init_attr {
    void                *qp_context; /* Associated context of the QP */
    struct ibv_cq        *send_cq;   /* CQ to be associated with the Send Queue (SQ) */
    struct ibv_cq        *recv_cq;   /* CQ to be associated with the Receive Queue (RQ) */
    struct ibv_srq        *srq;      /* SRQ handle if QP is to be associated with an SRQ, otherwise NULL */
    struct ibv_qp_cap     cap;       /* QP capabilities */
    enum ibv_qp_type      qp_type;   /* QP Transport Service Type: IBV_QPT_RC, IBV_QPT_UC, IBV_QPT_UD */
    int                  sq_sig_all; /* If set, each Work Request (WR) submitted to the SQ generates a completion */
};

struct ibv_qp_cap {
    uint32_t             max_send_wr; /* Requested max number of outstanding WRs in the SQ */
    uint32_t             max_recv_wr; /* Requested max number of outstanding WRs in the RQ */
    uint32_t             max_send_sge; /* Requested max number of scatter/gather (s/g) elements in a WR in the SQ */
    uint32_t             max_recv_sge; /* Requested max number of s/g elements in a WR in the RQ */
    uint32_t             max_inline_data; /* Requested max number of data (bytes) that can be posted inline */
};
```

The function **ibv_create_qp()** will update the *qp_init_attr->cap* struct with the actual QP values of the QP that was created; the values will be greater than or equal to the values requested.

ibv_destroy_qp() destroys the QP *qp*.

RETURN VALUE

ibv_create_qp() returns a pointer to the created QP, or NULL if the request fails. Check the QP number (*qp_num*) in the returned QP.

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

NOTES

ibv_create_qp() will fail if it is asked to create QP of a type other than **IBV_QPT_RC** or **IBV_QPT_UD** associated with an SRQ.

The attributes `max_recv_wr` and `max_recv_sge` are ignored by **ibv_create_qp()** if the QP is to be associated with an SRQ.

ibv_destroy_qp() fails if the QP is attached to a multicast group.

IBV_QPT_DRIVER does not represent a specific service and is used for vendor specific QP logic.

SEE ALSO

`ibv_alloc_pd(3)`, `ibv_modify_qp(3)`, `ibv_query_qp(3)`

AUTHORS

Dotan Barak <dotanba@gmail.com>