NAME

ibv_create_qp_ex, ibv_destroy_qp - create or destroy a queue pair (QP)

SYNOPSIS

#include <infiniband/verbs.h>

struct ibv_qp *ibv_create_qp_ex(struct ibv_context **context***,**
struct ibv_qp_init_attr_ex **qp_init_attr***);**

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

DESCRIPTION

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

```
struct ibv_qp_init_attr_ex {
        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_QP
        int                  sq_sig_all;   /* If set, each Work Request (WR) submitted to the SQ generates a completion
        uint32_t             comp_mask;  /* Identifies valid fields */
        struct ibv_pd       *pd;           /* PD to be associated with the QP */
        struct ibv_xrcd     *xrcd;                 /* XRC domain to be associated with the target QP */
        enum ibv_qp_create_flags create_flags;  /* Creation flags for this QP */
        uint16_t             max_tso_header; /* Maximum TSO header size */
        struct ibv_rwq_ind_table *rwq_ind_tbl;  /* Indirection table to be associated with the QP */
        struct ibv_rx_hash_conf  rx_hash_conf;  /* RX hash configuration to be used */
};

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
        uint32_t             max_recv_sge;   /* Requested max number of s/g elements in a WR in the SQ */
        uint32_t             max_inline_data;/* Requested max number of data (bytes) that can be posted inline to the S
};
enum ibv_qp_create_flags {
```

```
        IBV_QP_CREATE_BLOCK_SELF_MCAST_LB      = 1 << 1, /* Prevent self multicast loopback */
        IBV_QP_CREATE_SCATTER_FCS              = 1 << 8, /* FCS field will be scattered to host memory */
        IBV_QP_CREATE_CVLAN_STRIPPING          = 1 << 9, /* CVLAN field will be stripped from incoming p
};
struct ibv_rx_hash_conf {
        uint8_t         rx_hash_function;      /* RX hash function, use enum ibv_rx_hash_function_flags */
        uint8_t         rx_hash_key_len;       /* RX hash key length */
        uint8_t         *rx_hash_key;          /* RX hash key data */
        uint64_t        rx_hash_fields_mask;   /* RX fields that should participate in the hashing, use enum ibv_rx
};
```

The function **ibv_create_qp_ex()** will update the *qp_init_attr_ex*->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_ex() 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

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

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

SEE ALSO

ibv_alloc_pd(3), **ibv_modify_qp**(3), **ibv_query_qp**(3), **ibv_create_rwq_ind_table**(3)

AUTHORS

Yishai Hadas <yishaih@mellanox.com>