

NAME

`ibv_create_srq_ex`, `ibv_destroy_srq` – create or destroy a shared receive queue (SRQ)

SYNOPSIS

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

```
struct ibv_srq *ibv_create_srq_ex(struct ibv_context *context, struct
                                ibv_srq_init_attr_ex *srq_init_attr_ex);
```

```
int ibv_destroy_srq(struct ibv_srq *srq);
```

DESCRIPTION

ibv_create_srq_ex() creates a shared receive queue (SRQ) supporting both basic and xrc modes. The argument *srq_init_attr_ex* is an `ibv_srq_init_attr_ex` struct, as defined in `<infiniband/verbs.h>`.

```
struct ibv_srq_init_attr_ex {
    void                *srq_context; /* Associated context of the SRQ */
    struct ibv_srq_attr attr;          /* SRQ attributes */
    uint32_t            comp_mask;    /* Identifies valid fields */
    enum ibv_srq_type   srq_type;     /* Basic / XRC / tag matching */
    struct ibv_pd       *pd;          /* PD associated with the SRQ */
    struct ibv_xrcd     *xrcd;        /* XRC domain to associate with the SRQ */
    struct ibv_cq        *cq;         /* CQ to associate with the SRQ for XRC mode */
    struct ibv_tm_cap    tm_cap;      /* Tag matching attributes */
};

struct ibv_srq_attr {
    uint32_t            max_wr;        /* Requested max number of outstanding work requests (WRs) in the SRQ */
    uint32_t            max_sge;      /* Requested max number of scatter elements per WR */
    uint32_t            srq_limit;    /* The limit value of the SRQ */
};

struct ibv_tm_cap {
    uint32_t            max_num_tags; /* Tag matching list size */
    uint32_t            max_ops;      /* Number of outstanding tag list operations */
};
```

The function **ibv_create_srq_ex()** will update the *srq_init_attr_ex* struct with the original values of the SRQ that was created; the values of `max_wr` and `max_sge` will be greater than or equal to the values requested.

ibv_destroy_srq() destroys the SRQ *srq*.

RETURN VALUE

ibv_create_srq_ex() returns a pointer to the created SRQ, or NULL if the request fails.

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

NOTES

ibv_destroy_srq() fails if any queue pair is still associated with this SRQ.

SEE ALSO

ibv_alloc_pd(3), **ibv_modify_srq(3)**, **ibv_query_srq(3)**

AUTHORS

Yishai Hadas <yishaih@mellanox.com>