

**NAME**

mlx4dv\_init\_obj – Initialize mlx4 direct verbs object from ibv\_xxx structures

**SYNOPSIS**

```
#include <infiniband/mlx4dv.h>
```

```
int mlx4dv_init_obj(struct mlx4dv_obj *obj, uint64_t obj_type);
```

**DESCRIPTION**

**mlx4dv\_init\_obj()** This function will initialize mlx4dv\_xxx structs based on supplied type. The information for initialization is taken from ibv\_xx structs supplied as part of input.

Request information of CQ marks its owned by direct verbs for all consumer index related actions. The initialization type can be combination of several types together.

```
struct mlx4dv_qp {
    uint32_t    *rdb;
    uint32_t    *sdb;
    struct {
        uint32_t    wqe_cnt;
        int         wqe_shift;
        int         offset;
    } sq;
    struct {
        uint32_t    wqe_cnt;
        int         wqe_shift;
        int         offset;
    } rq;
    struct {
        void        *buf;
        size_t      length;
    } buf;
    uint64_t    comp_mask; /* Use enum mlx4dv_qp_comp_mask */
    off_t       uar_mmap_offset; /* If MLX4DV_QP_MASK_UAR_MMAP_OFFSET is set in comp_m
};

struct mlx4dv_cq {
    struct {
        void        *buf;
        size_t      length;
    } buf;
    uint32_t    cqe_cnt;
    uint32_t    cqn;
    uint32_t    *set_ci_db;
    uint32_t    *arm_db;
    int         arm_sn;
    int         cqe_size;
    uint64_t    comp_mask; /* Use enum mlx4dv_cq_comp_mask */
    void        *cq_uar;
};

struct mlx4dv_srq {
    struct {
        void        *buf;
        size_t      length;
    } buf;
```

```

        int      wqe_shift;
        int      head;
        int      tail;
        uint32_t  *db;
        uint64_t  comp_mask;
};

struct mlx4dv_rwq {
    __be32      *rdb;
    struct {
        uint32_t  wqe_cnt;
        int      wqe_shift;
        int      offset;
    } rq;
    struct {
        void      *buf;
        size_t    length;
    } buf;
    uint64_t     comp_mask;
};

struct mlx4dv_obj {
    struct {
        struct ibv_qp  *in;
        struct mlx4dv_qp *out;
    } qp;
    struct {
        struct ibv_cq  *in;
        struct mlx4dv_cq *out;
    } cq;
};

enum mlx4dv_obj_type {
    MLX4DV_OBJ_QP  = 1 << 0,
    MLX4DV_OBJ_CQ  = 1 << 1,
    MLX4DV_OBJ_SRQ = 1 << 2,
};

```

**RETURN VALUE**

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

**NOTES**

\* Compatibility masks (comp\_mask) are in/out fields.

**SEE ALSO**

**mlx4dv(7)**

**AUTHORS**

Maor Gottlieb <maorg@mellanox.com>