## NAME

mlx5dv_init_obj – Initialize mlx5 direct verbs object from ibv_xxx structures

## SYNOPSIS

**#include <infiniband/mlx5dv.h>**

**int mlx5dv_init_obj(struct mlx5dv_obj *obj, uint64_t obj_type);**

## DESCRIPTION

**mlx5dv_init_obj**() This function will initialize mlx5dv_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 mlx5dv_qp {
                uint32_t        *dbrec;
                struct {
                                void        *buf;
                                uint32_t     wqe_cnt;
                                uint32_t     stride;
                } sq;
                struct {
                                void        *buf;
                                uint32_t     wqe_cnt;
                                uint32_t     stride;
                } rq;
                struct {
                                void        *reg;
                                uint32_t     size;
                } bf;
                uint64_t        comp_mask;
                off_t                   uar_mmap_offset;
                uint32_t        tirn;
                uint32_t        tisn;
                uint32_t        rqn;
                uint32_t        sqn;
                uint64_t        tir_icm_address;
};

struct mlx5dv_cq {
                void            *buf;
                uint32_t         *dbrec;
                uint32_t         cqe_cnt;
                uint32_t         cqe_size;
                void            *cq_uar;
                uint32_t         cqn;
                uint64_t         comp_mask;
};

struct mlx5dv_srq {
                void            *buf;
                uint32_t         *dbrec;
                uint32_t         stride;
                uint32_t         head;
                uint32_t         tail;
                uint64_t        comp_mask;
                uint32_t        srqn;
};

struct mlx5dv_rwq {
                void        *buf;
                uint32_t    *dbrec;
                uint32_t    wqe_cnt;
                uint32_t    stride;
                uint64_t    comp_mask;
};

struct mlx5dv_dm {
                void             *buf;
                uint64_t length;
                uint64_t comp_mask;
                uint64_t remote_va;
};

struct mlx5dv_ah {
                struct mlx5_wqe_av    *av;
                uint64_t       comp_mask;
};

struct mlx5dv_pd {
                uint32_t       pdn;
                uint64_t       comp_mask;
};

struct mlx5dv_obj {
                struct {
                                struct ibv_qp         *in;
                                struct mlx5dv_qp      *out;
                } qp;
                struct {
                                struct ibv_cq         *in;
                                struct mlx5dv_cq      *out;
                } cq;
                struct {
                                struct ibv_srq        *in;
                                struct mlx5dv_srq     *out;
                } srq;
                struct {
                                struct ibv_wq         *in;
                                struct mlx5dv_rwq     *out;
                } rwq;
                struct {
                                struct ibv_dm               *in;
                                struct mlx5dv_dm            *out;
                } dm;
                struct {
                                struct ibv_ah               *in;
                                struct mlx5dv_ah *out;
                } ah;
                struct {
                                struct ibv_pd           *in;
                                struct mlx5dv_pd        *out;
                } pd;
        };

        enum mlx5dv_obj_type {
                MLX5DV_OBJ_QP   = 1 << 0,
                MLX5DV_OBJ_CQ   = 1 << 1,
                MLX5DV_OBJ_SRQ  = 1 << 2,
                MLX5DV_OBJ_RWQ  = 1 << 3,
                MLX5DV_OBJ_DM   = 1 << 4,
                MLX5DV_OBJ_AH   = 1 << 5,
                MLX5DV_OBJ_PD   = 1 << 6,
        };
```

## RETURN VALUE

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

## NOTES

* The information if doorbell is blueflame is based on mlx5dv_qp->bf->size, in case of 0 it's not a BF.
* Compatibility masks (comp_mask) are in/out fields.

## SEE ALSO

**mlx5dv**(7)

## AUTHORS

Leon Romanovsky <leonro@mellanox.com>