mlx5dv_devx_obj_create / destroy / modify /query / general(3)

NAME

mlx5dv_devx_obj_create – Creates a devx object

mlx5dv_devx_obj_destroy – Destroys a devx object

mlx5dv_devx_obj_modify – Modifies a devx object

mlx5dv_devx_obj_query – Queries a devx object

mlx5dv_devx_obj_query_async – Queries a devx object in an asynchronous mode

mlx5dv_devx_general_cmd – Issues a general command over the devx interface

SYNOPSIS

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

struct mlx5dv_devx_obj *
mlx5dv_devx_obj_create(struct ibv_context *context, const void *in, size_t
                       void *out, size_t outlen);
int mlx5dv_devx_obj_query(struct mlx5dv_devx_obj *obj, const void *in, size
                      void *out, size_t outlen);
int mlx5dv_devx_obj_query_async(struct mlx5dv_devx_obj *obj, const void *in
                       size_t inlen, size_t outlen,
                       uint64_t wr_id,
                       struct mlx5dv_devx_cmd_comp *cmd_comp);
int mlx5dv_devx_obj_modify(struct mlx5dv_devx_obj *obj, const void *in, siz
                       void *out, size_t outlen);
int mlx5dv_devx_obj_destroy(struct mlx5dv_devx_obj *obj);
int mlx5dv_devx_general_cmd(struct ibv_context *context, const void *in, si
                       void *out, size_t outlen);
```

DESCRIPTION

Create / destroy / modify / query a devx object, issue a general command over the devx interface.

The DEVX API enables direct access from the user space area to the mlx5 device driver by using the KABI mechanism. The main purpose is to make the user space driver as independent as possible from the kernel so that future device functionality and commands can be activated with minimal to none kernel changes.

A DEVX object represents some underlay firmware object, the input command to create it is some raw data given by the user application which should match the device specification. Upon successful creation the output buffer includes the raw data from the device according to its specification, this data can be used as part of related firmware commands to this object.

Once the DEVX object is created it can be queried/modified/destroyed by the matching mlx5dv_devx_obj_xxx() API. Both the input and the output for those APIs need to match the device specification as well.

The mlx5dv_devx_general_cmd() API enables issuing some general command which is not related to an object such as query device capabilities.

The mlx5dv_devx_obj_query_async() API is similar to the query object API, however, it runs asynchronously without blocking. The input includes an mlx5dv_devx_cmd_comp object and an identifier named 'wr_id' for this command. The response should be read upon success with the mlx5dv_devx_get_async_cmd_comp() API. The 'wr_id' that was supplied as an input is returned as part of the response to let application knows for which command the response is related to.

An application can gradually migrate to use DEVX according to its needs, it is not all or nothing. For example it can create an ibv_cq via ibv_create_cq() verb and then use the returned cqn to create a DEVX QP object by the mlx5dv_devx_obj_create() API which needs that cqn.

The above example can enable an application to create a QP with some driver specific attributes that are not exposed in the ibv_create_qp() API, in that case no user or kernel change may be needed at all as the command input reaches directly to the firmware.

mlx5dv_devx_obj_create / destroy / modify /query / general(3)

The expected users for the DEVX APIs are application that use the mlx5 DV APIs and are familiar with the device specification in both control and data path.

To successfully create a DEVX object and work on, a DEVX context must be created, this is done by the mlx5dv_open_device() API with the *MLX5DV_CONTEXT_FLAGS_DEVX* flag.

ARGUMENTS

context RDMA device context to create the action on.

in A buffer which contains the command's input data provided in a device specification format.

inlen The size of *in* buffer in bytes.

out A buffer which contains the command's output data according to the device specification format.

outlen The size of *out* buffer in bytes.

obj For query, modify, destroy: the devx object to work on.

wr_id The command identifier when working in asynchronous mode.

cmd_comp
The command completion object to read the response from in asynchronous mode.

RETURN VALUE

Upon success *mlx5dv_devx_create_obj* will return a new *struct mlx5dv_devx_obj* on error NULL will be returned and errno will be set.

Upon success query, modify, destroy, general commands, 0 is returned or the value of errno on a failure.

SEE ALSO

mlx5dv_open_device, **mlx5dv_devx_create_cmd_comp**, **mlx5dv_devx_get_async_cmd_comp**

#AUTHOR

Yishai Hadas <yishaih@mellanox.com>