NAME

ibv_modify_qp − modify the attributes of a queue pair (QP)

SYNOPSIS

#include <infiniband/verbs.h>

int ibv_modify_qp(struct ibv_qp **qp*, **struct ibv_qp_attr** **attr*,
  int *attr_mask***);**

DESCRIPTION

ibv_modify_qp() modifies the attributes of QP *qp* with the attributes in *attr* according to the mask *attr_mask*. The argument *attr* is an ibv_qp_attr struct, as defined in <infiniband/verbs.h>.

```
struct ibv_qp_attr {
        enum ibv_qp_state    qp_state;            /* Move the QP to this state */
        enum ibv_qp_state    cur_qp_state;        /* Assume this is the current QP state */
        enum ibv_mtu         path_mtu;            /* Path MTU (valid only for RC/UC QPs) */
        enum ibv_mig_state   path_mig_state;      /* Path migration state (valid if HCA supports APM) */
        uint32_t             qkey;                /* Q_Key for the QP (valid only for UD QPs) */
        uint32_t             rq_psn;              /* PSN for receive queue (valid only for RC/UC QPs) */
        uint32_t             sq_psn;              /* PSN for send queue (valid only for RC/UC QPs) */
        uint32_t             dest_qp_num;         /* Destination QP number (valid only for RC/UC QPs) */
        int                  qp_access_flags;     /* Mask of enabled remote access operations (valid only for RC/
        struct ibv_qp_cap    cap;                 /* QP capabilities (valid if HCA supports QP resizing) */
        struct ibv_ah_attr   ah_attr;             /* Primary path address vector (valid only for RC/UC QPs) */
        struct ibv_ah_attr   alt_ah_attr;         /* Alternate path address vector (valid only for RC/UC QPs) *
        uint16_t             pkey_index;          /* Primary P_Key index */
        uint16_t             alt_pkey_index;      /* Alternate P_Key index */
        uint8_t              en_sqd_async_notify; /* Enable SQD.drained async notification (Valid only if qp_
        uint8_t              sq_draining;         /* Is the QP draining? Irrelevant for ibv_modify_qp() */
        uint8_t              max_rd_atomic;       /* Number of outstanding RDMA reads & atomic operations
        uint8_t              max_dest_rd_atomic;  /* Number of responder resources for handling incoming R
        uint8_t              min_rnr_timer;       /* Minimum RNR NAK timer (valid only for RC QPs) */
        uint8_t              port_num;            /* Primary port number */
        uint8_t              timeout;             /* Local ack timeout for primary path (valid only for RC QPs) */
        uint8_t              retry_cnt;           /* Retry count (valid only for RC QPs) */
        uint8_t              rnr_retry;           /* RNR retry (valid only for RC QPs) */
        uint8_t              alt_port_num;        /* Alternate port number */
        uint8_t              alt_timeout;         /* Local ack timeout for alternate path (valid only for RC QPs) *
        uint32_t             rate_limit;          /* Rate limit in kbps for packet pacing */
};
```

For details on struct ibv_qp_cap see the description of **ibv_create_qp()**. For details on struct ibv_ah_attr see the description of **ibv_create_ah()**.

The argument *attr_mask* specifies the QP attributes to be modified. The argument is either 0 or the bitwise OR of one or more of the following flags:

IBV_QP_STATE Modify qp_state

IBV_QP_CUR_STATE Set cur_qp_state

IBV_QP_EN_SQD_ASYNC_NOTIFY Set en_sqd_async_notify

IBV_QP_ACCESS_FLAGS Set qp_access_flags

IBV_QP_PKEY_INDEX Set pkey_index

IBV_QP_PORT Set port_num

IBV_QP_QKEY Set qkey

IBV_QP_AV Set ah_attr

IBV_QP_PATH_MTU Set path_mtu

IBV_QP_TIMEOUT Set timeout

IBV_QP_RETRY_CNT Set retry_cnt

IBV_QP_RNR_RETRY Set rnr_retry

IBV_QP_RQ_PSN Set rq_psn

IBV_QP_MAX_QP_RD_ATOMIC Set max_rd_atomic

IBV_QP_ALT_PATH Set the alternative path via: alt_ah_attr, alt_pkey_index, alt_port_num, alt_timeout

IBV_QP_MIN_RNR_TIMER Set min_rnr_timer

IBV_QP_SQ_PSN Set sq_psn

IBV_QP_MAX_DEST_RD_ATOMIC Set max_dest_rd_atomic

IBV_QP_PATH_MIG_STATE Set path_mig_state

IBV_QP_CAP Set cap

IBV_QP_DEST_QPN Set dest_qp_num
IBV_QP_RATE_LIMIT Set rate_limit

RETURN VALUE

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

NOTES

If any of the modify attributes or the modify mask are invalid, none of the attributes will be modified (including the QP state).

Not all devices support resizing QPs. To check if a device supports it, check if the **IBV_DEVICE_RESIZE_MAX_WR** bit is set in the device capabilities flags.

Not all devices support alternate paths. To check if a device supports it, check if the **IBV_DEVICE_AUTO_PATH_MIG** bit is set in the device capabilities flags.

The following tables indicate for each QP Transport Service Type, the minimum list of attributes that must be changed upon transitioning QP state from: Reset --> Init --> RTR --> RTS.

For QP Transport Service Type **IBV_QPT_UD**:

Next state	Required attributes
Init	**IBV_QP_STATE, IBV_QP_PKEY_INDEX, IBV_QP_PORT, IBV_QP_QKEY**
RTR	**IBV_QP_STATE**
RTS	**IBV_QP_STATE, IBV_QP_SQ_PSN**

For QP Transport Service Type **IBV_QPT_UC**:

Next state	Required attributes
Init	**IBV_QP_STATE, IBV_QP_PKEY_INDEX, IBV_QP_PORT, IBV_QP_ACCESS_FLAGS**
RTR	**IBV_QP_STATE, IBV_QP_AV, IBV_QP_PATH_MTU, IBV_QP_DEST_QPN, IBV_QP_RQ_PSN**
RTS	**IBV_QP_STATE, IBV_QP_SQ_PSN**

For QP Transport Service Type **IBV_QPT_RC**:

Next state	Required attributes
Init	**IBV_QP_STATE, IBV_QP_PKEY_INDEX, IBV_QP_PORT, IBV_QP_ACCESS_FLAGS**
RTR	**IBV_QP_STATE, IBV_QP_AV, IBV_QP_PATH_MTU, IBV_QP_DEST_QPN, IBV_QP_RQ_PSN, IBV_QP_MAX_DEST_RD_ATOMIC, IBV_QP_MIN_RNR_TIMER**
RTS	**IBV_QP_STATE, IBV_QP_SQ_PSN, IBV_QP_MAX_QP_RD_ATOMIC, IBV_QP_RETRY_CNT, IBV_QP_RNR_RETRY, IBV_QP_TIMEOUT**

For QP Transport Service Type **IBV_QPT_RAW_PACKET**:

Next state	Required attributes
Init	**IBV_QP_STATE, IBV_QP_PORT**
RTR	**IBV_QP_STATE**
RTS	**IBV_QP_STATE**

If port flag IBV_QPF_GRH_REQUIRED is set then ah_attr and alt_ah_attr must be passed with definition of 'struct ibv_ah_attr { .is_global = 1; .grh = {...}; }'.

SEE ALSO

ibv_create_qp(3), **ibv_destroy_qp**(3), **ibv_query_qp**(3), **ibv_create_ah**(3)

AUTHORS

Dotan Barak <dotanba@gmail.com>