

NAME

`ibv_rereg_mr` - re-register a memory region (MR)

SYNOPSIS

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

```
int ibv_rereg_mr(struct ibv_mr *mr, int flags,
                 struct ibv_pd *pd, void *addr,
                 size_t length, int access);
```

DESCRIPTION

ibv_rereg_mr() Modifies the attributes of an existing memory region (MR) *mr*. Conceptually, this call performs the functions deregister memory region followed by register memory region. Where possible, resources are reused instead of deallocated and reallocated.

flags is a bit-mask used to indicate which of the following properties of the memory region are being modified. Flags should be a combination (bit field) of:

IBV_REREG_MR_CHANGE_TRANSLATION Change translation (location and length)

IBV_REREG_MR_CHANGE_PD Change protection domain

IBV_REREG_MR_CHANGE_ACCESS Change access flags

When **IBV_REREG_MR_CHANGE_PD** is used, *pd* represents the new PD this MR should be registered to.

When **IBV_REREG_MR_CHANGE_TRANSLATION** is used, *addr* represents the virtual address (user-space pointer) of the new MR, while *length* represents its length.

The access and other flags are represented in the field *access*. This field describes the desired memory protection attributes; it is either 0 or the bitwise OR of one or more of `ibv_access_flags`.

RETURN VALUE

ibv_rereg_mr() returns 0 on success, otherwise an error has occurred, *enum ibv_rereg_mr_err_code* represents the error as of below.

IBV_REREG_MR_ERR_INPUT - Old MR is valid, an input error was detected by libibverbs.

IBV_REREG_MR_ERR_DONT_FORK_NEW - Old MR is valid, failed via don't fork on new address range.

IBV_REREG_MR_ERR_DO_FORK_OLD - New MR is valid, failed via do fork on old address range.

IBV_REREG_MR_ERR_CMD - MR shouldn't be used, command error.

IBV_REREG_MR_ERR_CMD_AND_DO_FORK_NEW - MR shouldn't be used, command error, invalid fork state on new address range.

NOTES

Even on a failure, the user still needs to call `ibv_dereg_mr` on this MR.

SEE ALSO

`ibv_reg_mr(3)`, `ibv_dereg_mr(3)`,

AUTHORS

Matan Barak <matanb@mellanox.com>

Yishai Hadas <yishaih@mellanox.com>