



Linux Ubuntu 26.04 Manual Pages on command 'rtnetlink.3'

\$ man rtnetlink.3

rtnetlink(3) Library Functions Manual rtnetlink(3)

NAME

rtnetlink - macros to manipulate rtnetlink messages

LIBRARY

Standard C library (libc, -lc)

SYNOPSIS

```
#include <asm/types.h>

#include <linux/netlink.h>

#include <linux/rtnetlink.h>

#include <sys/socket.h>

rtnetlink_socket = socket(AF_NETLINK, int socket_type, NETLINK_ROUTE);

int RTA_OK(struct rtattr *rta, int size);

void *RTA_DATA(struct rtattr *rta);

unsigned int RTA_PAYLOAD(struct rtattr *rta);

struct rtattr *RTA_NEXT(struct rtattr *rta, unsigned int size);

unsigned int RTA_LENGTH(unsigned int size);

unsigned int RTA_SPACE(unsigned int size);
```

DESCRIPTION

All rtnetlink(7) messages consist of a netlink(7) message header and appended attributes.

The attributes should be manipulated only using the macros provided here.

RTA_OK(rta, size) returns true if rta points to a valid routing attribute; size is the

running size of the attribute buffer. When not true then you must assume there are no more at-

tributes in the message, even if size is nonzero.

RTA_DATA(rta) returns a pointer to the start of this attribute's data.

RTA_PAYLOAD(rta) returns the size of this attribute's data.

RTA_NEXT(rta, size) gets the next attribute after rta. Calling this macro will update size.

You should use RTA_OK to check the validity of the returned pointer.

RTA_LENGTH(size) returns the size which is required for size bytes of data plus the header.

RTA_SPACE(size) returns the amount of space which will be needed in a message with size bytes of data.

STANDARDS

Linux.

BUGS

This manual page is incomplete.

EXAMPLES

Creating a rtnetlink message to set the MTU of a device:

```
#include <linux/rtnetlink.h>

...

struct {
    struct nlmsghdr nh;
    struct ifinfomsg if;
    char          attrbuf[512];
} req;

struct rtattr *rta;

unsigned int mtu = 1000;

int rtnetlink_sk = socket(AF_NETLINK, SOCK_DGRAM, NETLINK_ROUTE);

memset(&req, 0, sizeof(req));

req.nh.nlmsg_len = NLMSG_LENGTH(sizeof(req.if));

req.nh.nlmsg_flags = NLM_F_REQUEST;

req.nh.nlmsg_type = RTM_NEWLINK;

req.if.if_family = AF_UNSPEC;

req.if.if_index = INTERFACE_INDEX;

req.if.if_change = 0xffffffff; /* ??? */

rta = (struct rtattr *) ((char *) &req +
                        NLMSG_ALIGN(req.nh.nlmsg_len));

rta->rta_type = IFLA_MTU;
```

```
rta->rta_len = RTA_LENGTH(sizeof(mtu));  
req.nh.nlmsg_len = NLMSG_ALIGN(req.nh.nlmsg_len) +  
    RTA_LENGTH(sizeof(mtu));  
memcpy(RTA_DATA(rta), &mtu, sizeof(mtu));  
send(rtnetlink_sk, &req, req.nh.nlmsg_len, 0);
```

SEE ALSO

netlink(3), netlink(7), rtnetlink(7)

Linux man-pages 6.17

2026-02-08

rtnetlink(3)