

**NAME**

`rmtstatus`, `rmtxstatus`, `_mtg2rmtg`, `_rmtg2mtg` - request `MTIOCGET` on a connection to a remote tape server (**-lrmt**)

**SYNOPSIS**

```
#include <schily/librmt.h>
```

```
#include <schily/rmtio.h>
```

```
int rmtstatus(int remfd, struct mtget *mtp);
```

```
int rmtxstatus(int remfd, struct rmtget *mtp);
```

```
void _rmtg2mtg(struct mtget *mtp, struct rmtget *rmtg);
```

```
int _mtg2rmtg(struct rmtget *rmtg, struct mtget *mtp);
```

**DESCRIPTION**

`rmtstatus()` and `rmtxstatus()`

perform a `MTIOCGET` request to the remote server, *remfd* is a file descriptor previously obtained from a call to `rmtgetconn(3)`, **struct mtget** is the local magnetic tape status structure, **struct rmtget** is the enhanced magnetic tape status structure from `librmt`. `rmtstatus(3)` and `rmtxstatus(3)` will fail if there was no previous successful `rmtopen(3)` before. `rmtstatus(3)` and `rmtxstatus(3)` take care of using **RMT protocol VERSION 1** if the remote side implements support for protocol version 1. The function `rmtstatus(3)` is outdated and should be avoided as the results in **struct mtget** will be the least common denominator of the local and remote variants of the structure. Use `rmtxstatus(3)` instead. The member `mt_xflags` in **struct rmtget** contains a bitmap that indicates which members of the structure contain valid values. See `mtio(4)` for more information.

`_rmtg2mtg()`

converts a **struct rmtget** into a **struct mtget**.

`_mtg2rmtg()`

converts a **struct mtget** into a **struct rmtget** and sets the member `mt_xflags` in **struct rmtget** to contain a bitmap that indicates which members of **struct mtget** are present in the local implementation.

**RETURN VALUES**

`rmtstatus()` and `rmtxstatus()`

return a value  $\geq 0$  if the remote `ioctl(f, MTIOCGET, struct mtget *)` succeeds.

**ERRORS**

**rmtstatus()** and **rmtxstatus()** return -1 on error and set **errno** to the **errno** value retrieved from the remote server.

**\_mtg2rmtg()**

returns -1 if no value from the local **struct mtget** could be converted to the abstract **struct rmtget**.

**EXAMPLES**

```

int    remfd;
char   *remfn;
char   host[256];
int    iosize = 10240; /* socket send/receive size to set up */
struct rmtget rmtg;

if ((remfn = rmtfilename(filename)) != NULL) {
    rmthostname(host, sizeof (host), filename);

    if ((remfd = rmtgetconn(host, iosize, 0)) < 0)
        comerrno(EX_BAD, "Cannot get connection to '%s'.\n",
                /* errno not valid !! */      host);
}

if (rmtopen(remfd, remfn, mode) < 0)
    comerr("Cannot open '%s'.\n", remfn);

if (rmtxstatus(remfd, &rmtg) < 0)
    comerr("Cannot retrieve magnetic tape status from '%s'.\n", remfn);

rmtclose(remfd);

```

**SEE ALSO**

**rmt(1)**, **rsh(1)**, **ssh(1)**, **rcmd(3)**, **rmtinit(3)**, **rmtdebug(3)**, **rmtrmt(3)**, **rmtrsh(3)**, **rmthostname(3)**, **rmtfilename(3)**, **rmtgetconn(3)**, **rmtopen(3)**, **rmtioctl(3)**, **rmtclose(3)**, **rmtread(3)**, **rmtwrite(3)**, **rmtseek(3)**, **rmtxstatus(3)**, **rmtstatus(3)**, **\_mtg2rmtg(3)**, **\_rmtg2mtg(3)**, **errmsgno(3)**, **mtio(4)**

**BUGS**

If local and remote **errno** values do not match, programs may get confused.

Mail other bugs and suggestions to [schilytools@mlists.in-berlin.de](mailto:schilytools@mlists.in-berlin.de) or open a ticket at

<https://codeberg.org/schilytools/schilytools/issues>.

The mailing list archive may be found at:

<https://mlists.in-berlin.de/mailman/listinfo/schilytools-mlists.in-berlin.de>.

## AUTHORS

**librmt** has been written in 1990 by Joerg Schilling. In 1995, support for **RMT VERSION 1** has been added. **librmt** is now maintained by the schilytools project authors.

## SOURCE DOWNLOAD

The source code for **librmt** is included in the **schilytools** project and may be retrieved from the **schilytools** project at Codeberg at

<https://codeberg.org/schilytools/schilytools>.

The download directory is

<https://codeberg.org/schilytools/schilytools/releases>.