

NAME

sctp_opt_info - get SCTP socket information

LIBRARY

Standard C Library (libc, -lc)

SYNOPSIS

```
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/sctp.h>
```

int

```
sctp_opt_info(int sd, sctp_assoc_t id, int opt, void *arg, socklen_t *size);
```

DESCRIPTION

The **sctp_opt_info()** call provides a multi-os compatible method for getting specific **getsockopt()** data where an association identification needs to be passed into the operating system. For FreeBSD a direct **getsockopt()** may be used, since FreeBSD has the ability to pass information into the operating system on a **getsockopt()** call. Other operating systems may not have this ability. For those who wish to write portable code amongst multiple operating systems this call should be used for the following SCTP socket options.

SCTP_RTOINFO

SCTP_ASSOCINFO

SCTP_PRIMARY_ADDR

SCTP_PEER_ADDR_PARAMS

SCTP_DEFAULT_SEND_PARAM

SCTP_MAX_SEG

SCTP_AUTH_ACTIVE_KEY

SCTP_DELAYED_SACK

SCTP_MAX_BURST

SCTP_CONTEXT

SCTP_EVENT

SCTP_DEFAULT_SNDINFO

SCTP_DEFAULT_PRINFO

SCTP_STATUS

SCTP_GET_PEER_ADDR_INFO

SCTP_PEER_AUTH_CHUNKS

SCTP_LOCAL_AUTH_CHUNKS

RETURN VALUES

The call returns 0 on success and -1 upon error.

ERRORS

The `sctp_opt_info()` function can return the following errors:

[EINVAL] The argument *arg* value was invalid.

[EOPNOTSUPP] The argument *opt* was not one of the above listed SCTP socket options.

[EBADF] The argument *s* is not a valid descriptor.

[ENOTSOCK] The argument *s* is not a socket.

SEE ALSO

getsockopt(2), sctp(4)

BUGS

Because the structure used for *arg* of the SCTP_MAX_BURST socket option has changed in FreeBSD 9.0 and higher, using SCTP_MAX_BURST as *opt* is only supported in FreeBSD 9.0 and higher.