

**NAME**

**sourcefilter** - advanced multicast group membership API

**SYNOPSIS**

```
#include <sys/socket.h>
```

```
#include <netinet/in.h>
```

*int*

```
getipv4sourcefilter(int s, struct in_addr interface, struct in_addr group, uint32_t *fmode,  
    uint32_t *numsrc, struct in_addr *slist);
```

*int*

```
getsourcefilter(int s, uint32_t interface, struct sockaddr *group, socklen_t grouplen, uint32_t *fmode,  
    uint32_t *numsrc, struct sockaddr_storage *slist);
```

*int*

```
setipv4sourcefilter(int s, struct in_addr interface, struct in_addr group, uint32_t fmode, uint32_t numsrc,  
    struct in_addr *slist);
```

*int*

```
setsourcefilter(int s, uint32_t interface, struct sockaddr *group, socklen_t grouplen, uint32_t fmode,  
    uint32_t numsrc, struct sockaddr_storage *slist);
```

**DESCRIPTION**

The **sourcefilter** functions implement the advanced, full-state multicast API defined in RFC 3678. An application may use these functions to atomically set and retrieve the multicast source address filters associated with a socket *s* and a multicast *group*.

The functions **getipv4sourcefilter**() and **getsourcefilter**() allow an application to discover the filter mode, and source filter entries, for an existing group membership.

The kernel will always return the number of source filter entries on the socket for that group in *\*numsrc*. If the *\*numsrc* argument is non-zero, the kernel will attempt to return up to *\*numsrc* filter entries in the array pointed to by *slist*. The *\*numsrc* argument may be set to 0, in which case the *slist* argument will be ignored.

For the **setipv4sourcefilter**() and **setsourcefilter**() functions, the *fmode* argument may be used to place the socket into inclusive or exclusive group membership modes, by using the MCAST\_INCLUDE or MCAST\_EXCLUDE constants respectively. The *numsrc* argument specifies the number of source entries in the *slist* array. If the *numsrc* argument has a value of 0, all source filters will be removed from

the socket. Removing all source filters from a membership which is in the `MCAST_INCLUDE` filter mode will cause the group to be left on that socket.

The protocol-independent function **setsourcefilter()** allows an application to join a multicast group on an interface which may not have an assigned protocol address, by passing its index for the *interface* argument.

Any changes made by these functions will be communicated to IGMPv3 and/or MLDv2 routers on the local network as appropriate. If no IGMPv3 or MLDv2 routers are present, changes in the source filter lists made by these functions will not cause state changes to be transmitted, with the exception of any change which causes a group to be joined or left. The kernel will continue to maintain the source filter state regardless of the IGMP or MLD version in use on the link.

## IMPLEMENTATION NOTES

The IPv4 specific versions of these functions are implemented in terms of the protocol-independent functions. Application writers are encouraged to use the protocol-independent functions for efficiency, and forwards compatibility with IPv6 networks.

For the protocol-independent functions **getsourcefilter()** and **setsourcefilter()**, the *grouplen* argument specifies the size of the structure pointed to by *group*. This is required in order to differentiate between different address families.

Currently FreeBSD does not support source address selection for the IPv4 protocol family, therefore the use of multicast APIs with an unnumbered IPv4 interface is *not recommended*. In all cases, the first assigned IPv4 address on the interface will be used as the source address of IGMP control traffic. If this address is removed or changed, the results are undefined.

## RETURN VALUES

The **getsourcefilter()**, **getipv4sourcefilter()**, **setsourcefilter()**, and **setipv4sourcefilter()** functions return the value 0 if successful; otherwise the value -1 is returned and the global variable *errno* is set to indicate the error.

## ERRORS

The **sourcefilter** functions may fail because of:

[EADDRNOTAVAIL]

The network interface which the interface argument refers to was not configured in the system, or the system is not a member of the group.

[EAFNOSUPPORT] The group and/or one or more of the *slist* arguments were of an address family

unsupported by the system, or the address family of the group and slist arguments were not identical.

[EINVAL] The group argument does not contain a multicast address. The fmode argument is invalid; it must be set to either MCAST\_INCLUDE or MCAST\_EXCLUDE. The numsrc or slist arguments do not specify a source list.

[ENOMEM] Insufficient memory was available to carry out the requested operation.

## SEE ALSO

ip(4), ip6(4), multicast(4), ifmcstat(8)

D. Thaler, B. Fenner, and B. Quinn, *Socket Interface Extensions for Multicast Source Filters*, RFC 3678, Jan 2004.

## HISTORY

The **sourcefilter** functions first appeared in FreeBSD 7.0.

## AUTHORS

Bruce M. Simpson <bms@FreeBSD.org>