

NAME

`ares_getsock` – get socket descriptors to wait on (deprecated)

SYNOPSIS

```
#include <ares.h>
```

```
int ares_getsock(const ares_channel_t *channel, ares_socket_t *socks,
                 int numsocks);
```

DESCRIPTION

The **ares_getsock** function retrieves the set of socket descriptors which the calling application should wait on for reading and/or writing for the processing of name service queries pending on the name service channel identified by *channel*. Socket descriptors will be set in the socket descriptor array pointed to by *socks*. *numsocks* is the size of the given array in number of ints.

This function can only return information up to 16 sockets. If more are in use, they are simply not reported back.

RETURN VALUES

ares_getsock returns a bitmask for what actions to wait for on the different sockets. The `ares.h` header file provides these convenience macros to extract the information appropriately:

```
#define ARES_GETSOCK_MAXNUM 16 /* ares_getsock() can return info about
                               this many sockets */
```

```
#define ARES_GETSOCK_READABLE(bits,num) (bits & (1<< (num)))
```

```
#define ARES_GETSOCK_WRITABLE(bits,num) (bits & (1<< ((num) +
```

`ARES_GETSOCK_MA`

NOTES

This function was added in c-ares 1.3.1 and deprecated in c-ares 1.20.0 due to the implementation of **ARES_OPT_MAX_UDP_QUERIES** which makes it likely to exceed 16 open file descriptors.

It is recommended to use **ARES_OPT_EVENT_THREAD** passed to `ares_init_options(3)` or to use socket state callbacks (**ARES_OPT_SOCK_STATE_CB**) registered via **ares_init_options(3)**.

SEE ALSO

`ares_init_options(3)`, `ares_timeout(3)`, `ares_fds(3)`, `ares_process(3)`