## NAME

ares_set_servers − Initialize an ares_channel name servers configuration

## SYNOPSIS

**#include <ares.h>**

**int ares_set_servers(ares_channel** *channel***, struct ares_addr_node ****servers***)**

## DESCRIPTION

The **ares_set_servers(3)** function initializes name servers configuration for the channel data identified by *channel*, from a *servers* pointer to a linked list of ares_addr_node structs holding name servers address data.

The name server linked list pointer argument may be the result of a previous call to **ares_get_servers(3)** or a linked list of ares_addr_node structs setup by other means.

This function replaces any potentially previously configured name servers with the ones given in the linked list. So, in order to configure a channel with more than one name server all the desired ones must be specified in a single list.

**ares_set_servers(3)** does not take ownership of the linked list argument. The caller is responsible to free the linked list when no longer needed.

This function is capable of handling IPv4 and IPv6 name server addresses simultaneously, rendering **ares_init_options(3)** with optmask **ARES_OPT_SERVERS** functionally obsolete except for IPv4-only name server usage.

## RETURN VALUES

**ares_set_servers(3)** may return any of the following values:

**ARES_SUCCESS**
The name servers configuration was successfuly initialized.

**ARES_ENOMEM**
The process's available memory was exhausted.

**ARES_ENODATA**
The channel data identified by *channel* was invalid.

**ARES_ENOTINITIALIZED**
c-ares library initialization not yet performed.

## SEE ALSO

**ares_get_servers**(3), **ares_init_options**(3), **ares_dup(3)**

## AVAILABILITY

ares_set_servers(3) was added in c-ares 1.7.1

## AUTHOR

Implementation of this function and associated library internals are based on code, comments and feedback provided November and December of 2008 by Daniel Stenberg, Gregor Jasny, Phil Blundell and Yang Tse, December 2009 by Cedric Bail, February 2010 by Jakub Hrozek. On March 2010 Yang Tse shuffled all the bits and this function popped out.