

## NAME

`ne_session_proxy`, `ne_session_socks_proxy`, `ne_session_system_proxy`, `ne_set_addrlist` - configure proxy servers

## SYNOPSIS

```
#include <ne_session.h>
```

```
void ne_session_proxy(ne_session *session, const char *hostname, unsigned int port);
```

```
void ne_session_system_proxy(ne_session *session, unsigned int flags);
```

```
void ne_session_socks_proxy(ne_session *session, enum ne_sock_sversion version,  
                           const char *hostname, unsigned int port,  
                           const char *username, const char *password);
```

```
void ne_set_addrlist(ne_session *session, const ne_inet_addr **addrlist, size_t count);
```

## DESCRIPTION

One (and no more than one) of the functions `ne_session_proxy`, `ne_session_system_proxy`, `ne_session_socks_proxy`, `ne_set_addrlist` can be used to configure a proxy server for a given session object. If more than one function is invoked for any given session object, only the last call has effect. If one of the functions is to be used, it must be used before the creation of any request object for the session.

### HTTP proxy specification

The `ne_session_proxy` function configures use of an HTTP proxy server for the session, the location of which is given by the `hostname` and `port` parameters. If the proxy requires authentication, `ne_set_proxy_auth` should be used.

### System proxy configuration

The `ne_session_system_proxy` function configures the session to use any proxy servers specified by the system configuration. Support for this function is platform-specific; if unsupported, the function has no effect.

### SOCKS proxy configuration

The `ne_session_socks_proxy` function configures the session to use a SOCKS proxy. The `version` indicates which version of the SOCKS protocol should be used. The `hostname` and `port` parameters specify the SOCKS proxy location. Note that a server with only an IPv6 address cannot be used with SOCKS v4 or v4A. The interpretation of the other arguments depends on the version specified:

**NE SOCK SOCKSV4** (version 4)

The *username* parameter must be non-NULL; the *password* parameter is ignored.

**NE SOCK SOCKSV4A** (version 4A)

The *username* parameter must be non-NULL; the *password* parameter is ignored.

**NE SOCK SOCKSV5** (version 5)

The *username* parameter may be NULL; if it is non-NULL, the *password* parameter must also be non-NULL; otherwise, it is ignored..

**Origin server address override**

The `ne_set_addrlist` function forces use of an address and port the a specified list when establishing a TCP connection, ignoring the "real" hostname and port identifying the origin server for the session (as passed to `ne_session_create`). The origin server's "real" hostname and port will still be used in the Host header in HTTP requests. When a connection is required, the library will iterate through the *addrlist* list, attempting to connect to the address *addrlist[0]* through to *addrlist[count-1]* in turn, until a connection can be established.

**RETURN VALUES**

None of the functions described here has a return value.

**EXAMPLES**

Create and destroy a session:

```
ne_session *sess;
sess = ne_session_create("http", "host.example.com", 80);
ne_session_proxy(sess, "proxy.example.com", 3128);
/* ... use sess ... */
ne_session_destroy(sess);
```

**SEE ALSO**

`ne_ssl_set_verify`, `ne_ssl_trust_cert`, `ne_sock_init`, `ne_set_session_flag`

**AUTHOR**

**Joe Orton**

Author.

**COPYRIGHT**