NAME

ne_sock_init, ne_sock_exit - perform library initialization

SYNOPSIS

#include <ne_socket.h>

int ne_sock_init(void);

void ne_sock_exit(void);

DESCRIPTION

In some platforms and configurations, neon may be using some socket or SSL libraries which require global initialization before use. To perform this initialization, the **ne_sock_init** function must be called before any other library functions are used.

Once all use of neon is complete, **ne_sock_exit** can be called to perform de-initialization of socket or SSL libraries, if necessary. Uses of **ne_sock_init** and **ne_sock_exit** are "reference counted"; if N calls to **ne_sock_init** are made, only the Nth call to **ne_sock_exit** will have effect.

ne_sock_init will set the disposition of the SIGPIPE signal to *ignored*. No change is made to the SIGPIPE disposition by **ne_sock_exit**.

Both the SSL libraries supported by neon -- OpenSSL and GnuTLS -- require callbacks to be registered to allow thread-safe use of SSL. These callbacks are stored as global variables and so their state persists for as long as the library in question is loaded into the process. If multiple users of the SSL library exist within the process, this can be problematic, particularly if one is dynamically loaded (and may subsequently be unloaded).

If neon is configured using the --enable-threadsafe-ssl flag, thread-safe SSL support will be enabled automatically, as covered in the following section. Otherwise, it is not safe to use neon with SSL in a multi-threaded process. The ne_has_support function can be used to determine whether neon is built to enable thread-safety support in the SSL library.

Thread-safe SSL with OpenSSL

neon follows two simple rules when dealing with the OpenSSL locking callbacks:

⊕

will set thread-safety locking callbacks if and only if no locking callbacks are already registered.

⊕

will unset the thread-safety locking callbacks if and only if the locking callbacks registered are those registered by **ne_sock_init**.

Applications and libraries should be able to co-operate to ensure that SSL use is always thread-safe if similar rules are always followed.

Thread-safe SSL with GnuTLS

The cryptography library used by GnuTLS, libgcrypt, only supports an initialization operation to register thread-safety callbacks. **ne_sock_init** will register the thread-safe locking callbacks on first use; **ne_sock_exit** cannot unregister them. If multiple users of GnuTLS are present within the process, it is unsafe to dynamically unload neon from the process if neon is configured with thread-safe SSL support enabled (since the callbacks would be left pointing at unmapped memory once neon is unloaded).

RETURN VALUE

ne_sock_init returns zero on success, or non-zero on error. If an error occurs, no further use of the neon library should be attempted.

SEE ALSO

neon(3), ne_has_support(3)

AUTHOR

Joe Orton <neon@lists.manyfish.co.uk>
       Author.

COPYRIGHT