

NAME

`ne_ssl_cert_read`, `ne_ssl_cert_write`, `ne_ssl_cert_import`, `ne_ssl_cert_export` - functions to read or write certificates to and from files or strings

SYNOPSIS

```
#include <ne_ssl.h>
```

```
ne_ssl_certificate *ne_ssl_cert_read(const char *filename);
```

```
int ne_ssl_cert_write(const ne_ssl_certificate *cert, const char *filename);
```

```
ne_ssl_certificate *ne_ssl_cert_import(const char *data);
```

```
char *ne_ssl_cert_export(const ne_ssl_certificate *cert);
```

DESCRIPTION

The `ne_ssl_cert_write` function writes a certificate to a file using the PEM encoding. The `ne_ssl_cert_export` function returns a base64-encoded NUL-terminated string representing the certificate. This string is malloc-allocated and should be destroyed using `free` by the caller.

The `ne_ssl_cert_read` function reads a certificate from a PEM-encoded file, and returns a certificate object. The `ne_ssl_cert_import` function returns a certificate object from a base64-encoded string, *data*, as returned by `ne_ssl_cert_export`. The certificate object returned by these functions should be destroyed using `ne_ssl_cert_free` after use.

RETURN VALUE

`ne_ssl_cert_read` returns NULL if a certificate could not be read from the file. `ne_ssl_cert_write` returns non-zero if the certificate could not be written to the file. `ne_ssl_cert_export` always returns a NUL-terminated string, and never NULL. `ne_ssl_cert_import` returns NULL if the string was not a valid base64-encoded certificate.

ENCODING FORMATS

The string produced by `ne_ssl_cert_export` is the base64 encoding of the DER representation of the certificate. The file written by `ne_ssl_cert_write` uses the PEM format: this is the base64 encoding of the DER representation with newlines every 64 characters, and start and end marker lines.

AUTHOR

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

Author.

COPYRIGHT