NAME

SSL_CTX_set_read_ahead, SSL_CTX_get_read_ahead, SSL_set_read_ahead, SSL_get_read_ahead, SSL_CTX_get_default_read_ahead - manage whether to read as many input bytes as possible

SYNOPSIS

```
#include <openssl/ssl.h>

void SSL_set_read_ahead(SSL *s, int yes);
int SSL_get_read_ahead(const SSL *s);

SSL_CTX_set_read_ahead(SSL_CTX *ctx, int yes);
long SSL_CTX_get_read_ahead(SSL_CTX *ctx);
long SSL_CTX_get_default_read_ahead(SSL_CTX *ctx);
```

DESCRIPTION

SSL_CTX_set_read_ahead() and *SSL_set_read_ahead()* set whether we should read as many input bytes as possible (for non-blocking reads) or not. For example if **x** bytes are currently required by OpenSSL, but **y** bytes are available from the underlying BIO (where **y** > **x**), then OpenSSL will read all **y** bytes into its buffer (providing that the buffer is large enough) if reading ahead is on, or **x** bytes otherwise. The parameter **yes** or **m** should be 0 to ensure reading ahead is off, or non zero otherwise. *SSL_CTX_set_default_read_ahead()* is identical to *SSL_CTX_set_read_ahead()*.

SSL_CTX_get_read_ahead() and *SSL_get_read_ahead()* indicate whether reading ahead has been set or not.

NOTES

These functions have no impact when used with DTLS. The return values for *SSL_CTX_get_read_head()* and *SSL_get_read_ahead()* are undefined for DTLS. Setting **read_ahead** can impact the behaviour of the *SSL_pending()* function (see *SSL_pending(3)*).

RETURN VALUES

SSL_get_read_ahead() and *SSL_CTX_get_read_ahead()* return 0 if reading ahead is off, and non zero otherwise.

SEE ALSO

ssl(3) , *SSL_pending(3)*

COPYRIGHT