

## NAME

SSL\_waiting\_for\_async, SSL\_get\_all\_async\_fds, SSL\_get\_changed\_async\_fds - manage asynchronous operations

## SYNOPSIS

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

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

```
int SSL_waiting_for_async(SSL *s);
```

```
int SSL_get_all_async_fds(SSL *s, OSSL_ASYNC_FD *fd, size_t *numfds);
```

```
int SSL_get_changed_async_fds(SSL *s, OSSL_ASYNC_FD *addfd, size_t *numaddfds,  
                             OSSL_ASYNC_FD *delfd, size_t *numdelfds);
```

## DESCRIPTION

**SSL\_waiting\_for\_async()** determines whether an SSL connection is currently waiting for asynchronous operations to complete (see the **SSL\_MODE\_ASYNC** mode in **SSL\_CTX\_set\_mode(3)**).

**SSL\_get\_all\_async\_fds()** returns a list of file descriptor which can be used in a call to **select()** or **poll()** to determine whether the current asynchronous operation has completed or not. A completed operation will result in data appearing as "read ready" on the file descriptor (no actual data should be read from the file descriptor). This function should only be called if the **SSL** object is currently waiting for asynchronous work to complete (i.e. **SSL\_ERROR\_WANT\_ASYNC** has been received - see **SSL\_get\_error(3)**). Typically the list will only contain one file descriptor. However, if multiple asynchronous capable engines are in use then more than one is possible. The number of file descriptors returned is stored in *\*numfds* and the file descriptors themselves are in *\*fds*. The *fds* parameter may be NULL in which case no file descriptors are returned but *\*numfds* is still populated. It is the callers responsibility to ensure sufficient memory is allocated at *\*fds* so typically this function is called twice (once with a NULL *fds* parameter and once without).

**SSL\_get\_changed\_async\_fds()** returns a list of the asynchronous file descriptors that have been added and a list that have been deleted since the last **SSL\_ERROR\_WANT\_ASYNC** was received (or since the **SSL** object was created if no **SSL\_ERROR\_WANT\_ASYNC** has been received). Similar to **SSL\_get\_all\_async\_fds()** it is the callers responsibility to ensure that *\*addfd* and *\*delfd* have sufficient memory allocated, although they may be NULL. The number of added fds and the number of deleted fds are stored in *\*numaddfds* and *\*numdelfds* respectively.

## RETURN VALUES

**SSL\_waiting\_for\_async()** will return 1 if the current SSL operation is waiting for an async operation to complete and 0 otherwise.

**SSL\_get\_all\_async\_fds()** and **SSL\_get\_changed\_async\_fds()** return 1 on success or 0 on error.

## NOTES

On Windows platforms the `<openssl/async.h>` header is dependent on some of the types customarily made available by including `<windows.h>`. The application developer is likely to require control over when the latter is included, commonly as one of the first included headers. Therefore, it is defined as an application developer's responsibility to include `<windows.h>` prior to `<openssl/async.h>`.

## SEE ALSO

`ssl(7)`, `SSL_get_error(3)`, `SSL_CTX_set_mode(3)`

## HISTORY

The `SSL_waiting_for_async()`, `SSL_get_all_async_fds()` and `SSL_get_changed_async_fds()` functions were added in OpenSSL 1.1.0.

## COPYRIGHT

Copyright 2016-2021 The OpenSSL Project Authors. All Rights Reserved.

Licensed under the Apache License 2.0 (the "License"). You may not use this file except in compliance with the License. You can obtain a copy in the file LICENSE in the source distribution or at <https://www.openssl.org/source/license.html>.