NAME

BIO_ADDR, BIO_ADDR_new, BIO_ADDR_clear, BIO_ADDR_free, BIO_ADDR_rawmake, BIO_ADDR_family, BIO_ADDR_rawaddress, BIO_ADDR_rawport, BIO_ADDR_hostname_string, BIO_ADDR_service_string, BIO_ADDR_path_string - BIO_ADDR routines

SYNOPSIS

```
#include <sys/types.h>
#include <openssl/bio.h>

typedef union bio_addr_st BIO_ADDR;

BIO_ADDR *BIO_ADDR_new(void);
void BIO_ADDR_free(BIO_ADDR *);
void BIO_ADDR_clear(BIO_ADDR *ap);
int BIO_ADDR_rawmake(BIO_ADDR *ap, int family,
                     const void *where, size_t wherelen, unsigned short port);
int BIO_ADDR_family(const BIO_ADDR *ap);
int BIO_ADDR_rawaddress(const BIO_ADDR *ap, void *p, size_t *l);
unsigned short BIO_ADDR_rawport(const BIO_ADDR *ap);
char *BIO_ADDR_hostname_string(const BIO_ADDR *ap, int numeric);
char *BIO_ADDR_service_string(const BIO_ADDR *ap, int numeric);
char *BIO_ADDR_path_string(const BIO_ADDR *ap);
```

DESCRIPTION

The **BIO_ADDR** type is a wrapper around all types of socket addresses that OpenSSL deals with, currently transparently supporting AF_INET, AF_INET6 and AF_UNIX according to what's available on the platform at hand.

BIO_ADDR_new() creates a new unfilled **BIO_ADDR**, to be used with routines that will fill it with information, such as **BIO_accept_ex()**.

BIO_ADDR_free() frees a **BIO_ADDR** created with **BIO_ADDR_new()**.

BIO_ADDR_clear() clears any data held within the provided **BIO_ADDR** and sets it back to an uninitialised state.

BIO_ADDR_rawmake() takes a protocol **family**, a byte array of size **wherelen** with an address in network byte order pointed at by **where** and a port number in network byte order in **port** (except for the **AF_UNIX** protocol family, where **port** is meaningless and therefore ignored) and populates the given **BIO_ADDR** with them. In case this creates a **AF_UNIX BIO_ADDR**, **wherelen** is expected to be the

length of the path string (not including the terminating NUL, such as the result of a call to **strlen()**). Read on about the addresses in "RAW ADDRESSES" below.

BIO_ADDR_family() returns the protocol family of the given **BIO_ADDR**. The possible non-error results are one of the constants AF_INET, AF_INET6 and AF_UNIX. It will also return AF_UNSPEC if the BIO_ADDR has not been initialised.

BIO_ADDR_rawaddress() will write the raw address of the given **BIO_ADDR** in the area pointed at by **p** if **p** is non-NULL, and will set ***l** to be the amount of bytes the raw address takes up if **l** is non-NULL. A technique to only find out the size of the address is a call with **p** set to **NULL**. The raw address will be in network byte order, most significant byte first. In case this is a **AF_UNIX** **BIO_ADDR**, **l** gets the length of the path string (not including the terminating NUL, such as the result of a call to **strlen()**). Read on about the addresses in "RAW ADDRESSES" below.

BIO_ADDR_rawport() returns the raw port of the given **BIO_ADDR**. The raw port will be in network byte order.

BIO_ADDR_hostname_string() returns a character string with the hostname of the given **BIO_ADDR**. If **numeric** is 1, the string will contain the numerical form of the address. This only works for **BIO_ADDR** of the protocol families AF_INET and AF_INET6. The returned string has been allocated on the heap and must be freed with **OPENSSL_free()**.

BIO_ADDR_service_string() returns a character string with the service name of the port of the given **BIO_ADDR**. If **numeric** is 1, the string will contain the port number. This only works for **BIO_ADDR** of the protocol families AF_INET and AF_INET6. The returned string has been allocated on the heap and must be freed with **OPENSSL_free()**.

BIO_ADDR_path_string() returns a character string with the path of the given **BIO_ADDR**. This only works for **BIO_ADDR** of the protocol family AF_UNIX. The returned string has been allocated on the heap and must be freed with **OPENSSL_free()**.

RAW ADDRESSES

Both **BIO_ADDR_rawmake()** and **BIO_ADDR_rawaddress()** take a pointer to a network byte order address of a specific site. Internally, those are treated as a pointer to **struct in_addr** (for **AF_INET**), **struct in6_addr** (for **AF_INET6**) or **char *** (for **AF_UNIX**), all depending on the protocol family the address is for.

RETURN VALUES

The string producing functions **BIO_ADDR_hostname_string()**, **BIO_ADDR_service_string()** and **BIO_ADDR_path_string()** will return **NULL** on error and leave an error indication on the OpenSSL

error stack.

All other functions described here return 0 or **NULL** when the information they should return isn't available.

SEE ALSO

BIO_connect(3), **BIO_s_connect**(3)

COPYRIGHT