NAME

ASN1_STRING_dup, ASN1_STRING_cmp, ASN1_STRING_set, ASN1_STRING_length, ASN1_STRING_type, ASN1_STRING_get0_data, ASN1_STRING_data, ASN1_STRING_to_UTF8 − ASN1_STRING utility functions

SYNOPSIS

```
#include <openssl/asn1.h>

int ASN1_STRING_length(ASN1_STRING *x);
const unsigned char *ASN1_STRING_get0_data(const ASN1_STRING *x);
unsigned char *ASN1_STRING_data(ASN1_STRING *x);

ASN1_STRING *ASN1_STRING_dup(const ASN1_STRING *a);

int ASN1_STRING_cmp(ASN1_STRING *a, ASN1_STRING *b);

int ASN1_STRING_set(ASN1_STRING *str, const void *data, int len);

int ASN1_STRING_type(const ASN1_STRING *x);

int ASN1_STRING_to_UTF8(unsigned char **out, const ASN1_STRING *in);
```

DESCRIPTION

These functions allow an **ASN1_STRING** structure to be manipulated.

ASN1_STRING_length() returns the length of the content of *x*. *x* **MUST NOT** be NULL.

ASN1_STRING_get0_data() returns an internal pointer to the data of *x*. Since this is an internal pointer it should **not** be freed or modified in any way. *x* **MUST NOT** be NULL.

ASN1_STRING_data() is similar to **ASN1_STRING_get0_data()** except the returned value is not constant. This function is deprecated: applications should use **ASN1_STRING_get0_data()** instead.

ASN1_STRING_dup() returns a copy of the structure *a*.

ASN1_STRING_cmp() compares *a* and *b* returning 0 if the two are identical. The string types and content are compared.

ASN1_STRING_set() sets the data of string *str* to the buffer *data* or length *len*. The supplied data is copied. If *len* is −1 then the length is determined by strlen(data).

ASN1_STRING_type() returns the type of *x*, using standard constants such as **V_ASN1_OCTET_STRING**.

ASN1_STRING_to_UTF8() converts the string *in* to UTF8 format, the converted data is allocated in a buffer in **out*. The length of *out* is returned or a negative error code. The buffer **out* should be freed using **OPENSSL_free()**.

NOTES

Almost all ASN1 types in OpenSSL are represented as an **ASN1_STRING** structure. Other types such as **ASN1_OCTET_STRING** are simply typedef'ed to **ASN1_STRING** and the functions call the **ASN1_STRING** equivalents. **ASN1_STRING** is also used for some **CHOICE** types which consist entirely of primitive string types such as **DirectoryString** and **Time**.

These functions should **not** be used to examine or modify **ASN1_INTEGER** or **ASN1_ENUMERATED** types: the relevant **INTEGER** or **ENUMERATED** utility functions should be used instead.

In general it cannot be assumed that the data returned by **ASN1_STRING_data()** is null terminated or does not contain embedded nulls. The actual format of the data will depend on the actual string type itself: for example for an IA5String the data will be ASCII, for a BMPString two bytes per character in big endian format, and for a UTF8String it will be in UTF8 format.

Similar care should be take to ensure the data is in the correct format when calling **ASN1_STRING_set()**.

RETURN VALUES

ASN1_STRING_length() returns the length of the content of *x*.

ASN1_STRING_get0_data() and **ASN1_STRING_data()** return an internal pointer to the data of *x*.

ASN1_STRING_dup() returns a valid **ASN1_STRING** structure or NULL if an error occurred.

ASN1_STRING_cmp() returns an integer greater than, equal to, or less than 0, according to whether *a* is greater than, equal to, or less than *b*.

ASN1_STRING_set() returns 1 on success or 0 on error.

ASN1_STRING_type() returns the type of *x*.

ASN1_STRING_to_UTF8() returns the number of bytes in output string *out* or a negative value if an error occurred.

SEE ALSO

ERR_get_error (3)

COPYRIGHT

Copyright 2002−2020 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>.