

NAME

X509_NAME_add_entry_by_txt, X509_NAME_add_entry_by_OBJ,
 X509_NAME_add_entry_by_NID, X509_NAME_add_entry, X509_NAME_delete_entry -
 X509_NAME modification functions

SYNOPSIS

```
#include <openssl/x509.h>
```

```
int X509_NAME_add_entry_by_txt(X509_NAME *name, const char *field, int type,  

    const unsigned char *bytes, int len, int loc, int set);
```

```
int X509_NAME_add_entry_by_OBJ(X509_NAME *name, const ASN1_OBJECT *obj, int type,  

    const unsigned char *bytes, int len, int loc, int set);
```

```
int X509_NAME_add_entry_by_NID(X509_NAME *name, int nid, int type,  

    const unsigned char *bytes, int len, int loc, int set);
```

```
int X509_NAME_add_entry(X509_NAME *name, const X509_NAME_ENTRY *ne, int loc, int set);
```

```
X509_NAME_ENTRY *X509_NAME_delete_entry(X509_NAME *name, int loc);
```

DESCRIPTION

X509_NAME_add_entry_by_txt(), **X509_NAME_add_entry_by_OBJ()** and **X509_NAME_add_entry_by_NID()** add a field whose name is defined by a string **field**, an object **obj** or a NID **nid** respectively. The field value to be added is in **bytes** of length **len**. If **len** is -1 then the field length is calculated internally using `strlen(bytes)`.

The type of field is determined by **type** which can either be a definition of the type of **bytes** (such as **MBSTRING_ASC**) or a standard ASN1 type (such as **V_ASN1_IA5STRING**). The new entry is added to a position determined by **loc** and **set**.

X509_NAME_add_entry() adds a copy of **X509_NAME_ENTRY** structure **ne** to **name**. The new entry is added to a position determined by **loc** and **set**. Since a copy of **ne** is added **ne** must be freed up after the call.

X509_NAME_delete_entry() deletes an entry from **name** at position **loc**. The deleted entry is returned and must be freed up.

NOTES

The use of string types such as **MBSTRING_ASC** or **MBSTRING_UTF8** is strongly recommended for

the **type** parameter. This allows the internal code to correctly determine the type of the field and to apply length checks according to the relevant standards. This is done using **ASN1_STRING_set_by_NID()**.

If instead an ASN1 type is used no checks are performed and the supplied data in **bytes** is used directly.

In **X509_NAME_add_entry_by_txt()** the **field** string represents the field name using **OBJ_txt2obj(field, 0)**.

The **loc** and **set** parameters determine where a new entry should be added. For almost all applications **loc** can be set to -1 and **set** to 0. This adds a new entry to the end of **name** as a single valued RelativeDistinguishedName (RDN).

loc actually determines the index where the new entry is inserted: if it is -1 it is appended.

set determines how the new type is added. If it is zero a new RDN is created.

If **set** is -1 or 1 it is added as a new set member to the previous or next RDN structure, respectively. This will then become part of a multi-valued RDN (containing a set of AVAs). Since multi-valued RDNs are very rarely used **set** typically will be zero.

RETURN VALUES

X509_NAME_add_entry_by_txt(), **X509_NAME_add_entry_by_OBJ()**, **X509_NAME_add_entry_by_NID()** and **X509_NAME_add_entry()** return 1 for success or 0 if an error occurred.

X509_NAME_delete_entry() returns either the deleted **X509_NAME_ENTRY** structure or **NULL** if an error occurred.

EXAMPLES

Create an **X509_NAME** structure:

```
"C=UK, O=Disorganized Organization, CN=Joe Bloggs"
```

```
X509_NAME *nm;
```

```
nm = X509_NAME_new();
```

```
if (nm == NULL)
```

```
    /* Some error */
```

```
if (!X509_NAME_add_entry_by_txt(nm, "C", MBSTRING_ASC,
```

X509_NAME_ADD_ENTRY_BY_TXT(3oss) OpenSSL X509_NAME_ADD_ENTRY_BY_TXT(3oss)

```
        "UK", -1, -1, 0))
/* Error */
if (!X509_NAME_add_entry_by_txt(nm, "O", MBSTRING_ASC,
        "Disorganized Organization", -1, -1, 0))
/* Error */
if (!X509_NAME_add_entry_by_txt(nm, "CN", MBSTRING_ASC,
        "Joe Bloggs", -1, -1, 0))
/* Error */
```

BUGS

type can still be set to **V_ASN1_APP_CHOOSE** to use a different algorithm to determine field types. Since this form does not understand multicharacter types, performs no length checks and can result in invalid field types its use is strongly discouraged.

SEE ALSO

ERR_get_error(3), **d2i_X509_NAME(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>.