

**NAME**

X509\_NAME\_get\_index\_by\_NID, X509\_NAME\_get\_index\_by\_OBJ, X509\_NAME\_get\_entry, X509\_NAME\_entry\_count, X509\_NAME\_get\_text\_by\_NID, X509\_NAME\_get\_text\_by\_OBJ - X509\_NAME lookup and enumeration functions

**SYNOPSIS**

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

```
int X509_NAME_get_index_by_NID(const X509_NAME *name, int nid, int lastpos);
```

```
int X509_NAME_get_index_by_OBJ(const X509_NAME *name,
                               const ASN1_OBJECT *obj, int lastpos);
```

```
int X509_NAME_entry_count(const X509_NAME *name);
```

```
X509_NAME_ENTRY *X509_NAME_get_entry(const X509_NAME *name, int loc);
```

```
int X509_NAME_get_text_by_NID(const X509_NAME *name, int nid,
                              char *buf, int len);
```

```
int X509_NAME_get_text_by_OBJ(const X509_NAME *name, const ASN1_OBJECT *obj,
                              char *buf, int len);
```

**DESCRIPTION**

These functions allow an **X509\_NAME** structure to be examined. The **X509\_NAME** structure is the same as the **Name** type defined in RFC2459 (and elsewhere) and used for example in certificate subject and issuer names.

**X509\_NAME\_get\_index\_by\_NID()** and **X509\_NAME\_get\_index\_by\_OBJ()** retrieve the next index matching **nid** or **obj** after **lastpos**. **lastpos** should initially be set to -1. If there are no more entries -1 is returned. If **nid** is invalid (doesn't correspond to a valid OID) then -2 is returned.

**X509\_NAME\_entry\_count()** returns the total number of entries in **name**.

**X509\_NAME\_get\_entry()** retrieves the **X509\_NAME\_ENTRY** from **name** corresponding to index **loc**. Acceptable values for **loc** run from 0 to (X509\_NAME\_entry\_count(name) - 1). The value returned is an internal pointer which must not be freed.

**X509\_NAME\_get\_text\_by\_NID()**, **X509\_NAME\_get\_text\_by\_OBJ()** retrieve the "text" from the first entry in **name** which matches **nid** or **obj**, if no such entry exists -1 is returned. At most **len** bytes will be written and the text written to **buf** will be null terminated. The length of the output string written is returned excluding the terminating null. If **buf** is <NULL> then the amount of space needed in **buf** (excluding the final null) is returned.

**NOTES**

**X509\_NAME\_get\_text\_by\_NID()** and **X509\_NAME\_get\_text\_by\_OBJ()** should be considered deprecated because they have various limitations which make them of minimal use in practice. They can only find the first matching entry and will copy the contents of the field verbatim: this can be highly confusing if the target is a multicharacter string type like a BMPString or a UTF8String.

For a more general solution **X509\_NAME\_get\_index\_by\_NID()** or **X509\_NAME\_get\_index\_by\_OBJ()** should be used followed by **X509\_NAME\_get\_entry()** on any matching indices and then the various **X509\_NAME\_ENTRY** utility functions on the result.

The list of all relevant **NID\_\*** and **OBJ\_\*** codes can be found in the source code header files `<openssl/obj_mac.h>` and/or `<openssl/objects.h>`.

Applications which could pass invalid NIDs to **X509\_NAME\_get\_index\_by\_NID()** should check for the return value of -2. Alternatively the NID validity can be determined first by checking **OBJ\_nid2obj(nid)** is not NULL.

**RETURN VALUES**

**X509\_NAME\_get\_index\_by\_NID()** and **X509\_NAME\_get\_index\_by\_OBJ()** return the index of the next matching entry or -1 if not found. **X509\_NAME\_get\_index\_by\_NID()** can also return -2 if the supplied NID is invalid.

**X509\_NAME\_entry\_count()** returns the total number of entries, and 0 for failure.

**X509\_NAME\_get\_entry()** returns an **X509\_NAME** pointer to the requested entry or **NULL** if the index is invalid.

**EXAMPLES**

Process all entries:

```
int i;
X509_NAME_ENTRY *e;

for (i = 0; i < X509_NAME_entry_count(nm); i++) {
    e = X509_NAME_get_entry(nm, i);
    /* Do something with e */
}
```

Process all commonName entries:

```
int lastpos = -1;
X509_NAME_ENTRY *e;

for (;;) {
    lastpos = X509_NAME_get_index_by_NID(nm, NID_commonName, lastpos);
    if (lastpos == -1)
        break;
    e = X509_NAME_get_entry(nm, lastpos);
    /* Do something with e */
}
```

### SEE ALSO

**ERR\_get\_error(3)**, **d2i\_X509\_NAME(3)**

### COPYRIGHT

Copyright 2002-2021 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>.