



*Full credit is given to the above companies including the OS that this PDF file was generated!*

### ***Rocky Enterprise Linux 9.2 Manual Pages on command 'X509\_NAME\_print\_ex\_fp.3ossl'***

***\$ man X509\_NAME\_print\_ex\_fp.3ossl***

X509\_NAME\_PRINT\_EX(3ossl)      OpenSSL      X509\_NAME\_PRINT\_EX(3ossl)

#### NAME

X509\_NAME\_print\_ex, X509\_NAME\_print\_ex\_fp, X509\_NAME\_print,  
X509\_NAME\_online - X509\_NAME printing routines

#### SYNOPSIS

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

int X509_NAME_print_ex(BIO *out, const X509_NAME *nm,
                      int indent, unsigned long flags);

int X509_NAME_print_ex_fp(FILE *fp, const X509_NAME *nm,
                          int indent, unsigned long flags);

char *X509_NAME_online(const X509_NAME *a, char *buf, int size);

int X509_NAME_print(BIO *bp, const X509_NAME *name, int obase);
```

#### DESCRIPTION

X509\_NAME\_print\_ex() prints a human readable version of nm to BIO out.

Each line (for multiline formats) is indented by indent spaces. The output format can be extensively customised by use of the flags parameter.

X509\_NAME\_print\_ex\_fp() is identical to X509\_NAME\_print\_ex() except the

output is written to FILE pointer fp.

X509\_NAME\_oneline() prints an ASCII version of a to buf. This supports multi-valued RDNs and escapes / and + characters in values. If buf is NULL then a buffer is dynamically allocated and returned, and size is ignored. Otherwise, at most size bytes will be written, including the ending '\0', and buf is returned.

X509\_NAME\_print() prints out name to bp indenting each line by obase characters. Multiple lines are used if the output (including indent) exceeds 80 characters.

## NOTES

The functions X509\_NAME\_oneline() and X509\_NAME\_print() produce a non standard output form, they don't handle multi-character fields and have various quirks and inconsistencies. Their use is strongly discouraged in new applications and they could be deprecated in a future release.

Although there are a large number of possible flags for most purposes XN\_FLAG\_ONELINE, XN\_FLAG\_MULTILINE or XN\_FLAG\_RFC2253 will suffice. As noted on the ASN1\_STRING\_print\_ex(3) manual page for UTF8 terminals the ASN1\_STRFLGS\_ESC\_MSB should be unset: so for example XN\_FLAG\_ONELINE & ~ASN1\_STRFLGS\_ESC\_MSB would be used.

The complete set of the flags supported by X509\_NAME\_print\_ex() is listed below.

Several options can be ored together.

The options XN\_FLAG\_SEP\_COMMA\_PLUS, XN\_FLAG\_SEP\_CPLUS\_SPC, XN\_FLAG\_SEP\_SPLUS\_SPC and XN\_FLAG\_SEP\_MULTILINE determine the field separators to use. Two distinct separators are used between distinct RelativeDistinguishedName components and separate values in the same RDN for a multi-valued RDN. Multi-valued RDNs are currently very rare so the second separator will hardly ever be used.

XN\_FLAG\_SEP\_COMMA\_PLUS uses comma and plus as separators.

XN\_FLAG\_SEP\_CPLUS\_SPC uses comma and plus with spaces: this is more readable than plain comma and plus. XN\_FLAG\_SEP\_SPLUS\_SPC uses spaced semicolon and plus. XN\_FLAG\_SEP\_MULTILINE uses spaced newline and plus respectively.

If XN\_FLAG\_DN\_REV is set the whole DN is printed in reversed order.

The fields XN\_FLAG\_FN\_SN, XN\_FLAG\_FN\_LN, XN\_FLAG\_FN\_OID, XN\_FLAG\_FN\_NONE determine how a field name is displayed. It will use the short name (e.g. CN) the long name (e.g. commonName) always use OID numerical form (normally OIDs are only used if the field name is not recognised) and no field name respectively.

If XN\_FLAG\_SPC\_EQ is set then spaces will be placed around the '=' character separating field names and values.

If XN\_FLAG\_DUMP\_UNKNOWN\_FIELDS is set then the encoding of unknown fields is printed instead of the values.

If XN\_FLAG\_FN\_ALIGN is set then field names are padded to 20 characters: this is only of use for multiline format.

Additionally all the options supported by ASN1\_STRING\_print\_ex() can be used to control how each field value is displayed.

In addition a number options can be set for commonly used formats.

XN\_FLAG\_RFC2253 sets options which produce an output compatible with RFC2253. It is equivalent to:

```
"ASN1_STRFLGS_RFC2253 | XN_FLAG_SEP_COMMA_PLUS | XN_FLAG_DN_REV  
| XN_FLAG_FN_SN | XN_FLAG_DUMP_UNKNOWN_FIELDS"
```

XN\_FLAG\_ONELINE is a more readable one line format which is the same as:

```
"ASN1_STRFLGS_RFC2253 | ASN1_STRFLGS_ESC_QUOTE | XN_FLAG_SEP_CPLUS_SPC  
| XN_FLAG_SPC_EQ | XN_FLAG_FN_SN"
```

XN\_FLAG\_MULTILINE is a multiline format which is the same as:

```
"ASN1_STRFLGS_ESC_CTRL | ASN1_STRFLGS_ESC_MSB | XN_FLAG_SEP_MULTILINE  
| XN_FLAG_SPC_EQ | XN_FLAG_FN_LN | XN_FLAG_FN_ALIGN"
```

XN\_FLAG\_COMPAT uses a format identical to X509\_NAME\_print(): in fact it calls X509\_NAME\_print() internally.

## RETURN VALUES

X509\_NAME\_oneline() returns a valid string on success or NULL on error.

X509\_NAME\_print() returns 1 on success or 0 on error.

X509\_NAME\_print\_ex() and X509\_NAME\_print\_ex\_fp() return 1 on success or 0 on error if the XN\_FLAG\_COMPAT is set, which is the same as

X509\_NAME\_print(). Otherwise, it returns -1 on error or other values on success.

#### SEE ALSO

ASN1\_STRING\_print\_ex(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>.

3.0.7                    2023-07-13        X509\_NAME\_PRINT\_EX(3openssl)