asn1_write_value(3)

NAME

asn1_write_value - API function

SYNOPSIS

#include <libtasn1.h>

int asn1_write_value(asn1_nodenode_root**, const char ***name**, const void ***ivalue**, int**len**);**

ARGUMENTS

asn1_node node_root pointer to a structure

const char * name the name of the element inside the structure that you want to set.

const void * ivalue vector used to specify the value to set. If len is >0, VALUE must be a two's complement form integer. if len=0 *VALUE must be a null terminated string with an integer value.

int len number of bytes of *value to use to set the value: value[0]..value[len-1] or 0 if value is a null terminated string

DESCRIPTION

Set the value of one element inside a structure.

If an element is OPTIONAL and you want to delete it, you must use the value=NULL and len=0. Using “pkix.asn”:

result=asn1_write_value(cert, “tbsCertificate.issuerUniqueID”, NULL, 0);

Description for each type:

INTEGER

VALUE must contain a two’s complement form integer.

value[0]=0xFF , len=1 -> integer=-1. value[0]=0xFF value[1]=0xFF , len=2 -> integer=-1. value[0]=0x01 , len=1 -> integer= 1. value[0]=0x00 value[1]=0x01 , len=2 -> integer= 1. value=“123” , len=0 -> integer= 123.

ENUMERATED

As INTEGER (but only with not negative numbers).

BOOLEAN

VALUE must be the null terminated string “TRUE” or “FALSE” and LEN != 0.

value=“TRUE” , len=1 -> boolean=TRUE. value=“FALSE” , len=1 -> boolean=FALSE.

OBJECT IDENTIFIER: VALUE must be a null terminated string with each number separated by a dot (e.g. “1.2.3.543.1”). LEN != 0.

value=“1 2 840 10040 4 3” , len=1 -> OID=dsa-with-sha.

UTCTIME

VALUE must be a null terminated string in one of these formats: “YYMMDDhhmmssZ”, “YYMMDDhhmmssZ”, “YYMMDDhhmmss+hh’mm”’, “YYMMDDhhmmss-hh’mm”’, “YYMMDDhhmm+hh’mm”’, or “YYMMDDhhmm-hh’mm”’. LEN != 0.

value=“9801011200Z” , len=1 -> time=Jannuary 1st, 1998 at 12h 00m Greenwich Mean Time

GENERALIZEDTIME

VALUE must be in one of this format: “YYYYMMDDhhmmss.sZ”, “YYYYMMDDhhmmss.sZ”, “YYYYMMDDhhmmss.s+hh’mm”’, “YYYYMMDDhhmmss.s-hh’mm”’, “YYYYMMDDhhmm+hh’mm”’, or “YYYYMMDDhhmm-hh’mm”’ where ss.s indicates the seconds with any precision like “10.1” or “01.02”. LEN != 0

value=“2001010112001.12-0700” , len=1 -> time=Jannuary 1st, 2001 at 12h 00m 01.12s Pacific Daylight Time

OCTET STRING: VALUE contains the octet string and LEN is the number of octets.

value=“shx01shx02shx03” , len=3 -> three bytes octet string

GENERALSTRING

VALUE contains the generalstring and LEN is the number of octets.

value=“shx01shx02shx03” , len=3 -> three bytes generalstring

BIT STRING: VALUE contains the bit string organized by bytes and LEN is the number of bits.

value=“shxCF” , len=6 -> bit string=“110011” (six bits)

CHOICE

if NAME indicates a choice type, VALUE must specify one of the alternatives with a null terminated string. LEN != 0. Using “pkix.asn”

result=asn1_write_value(cert, “certificate1.tbsCertificate.subject”, “rdnSequence”, 1);

ANY

VALUE indicates the der encoding of a structure. LEN != 0.

SEQUENCE OF: VALUE must be the null terminated string “NEW” and LEN != 0. With this instruction another element is appended in the sequence. The name of this element will be “?1” if it’s the first one, “?2” for the second and so on.

Using “pkix.asn”

result=asn1_write_value(cert, “certificate1.tbsCertificate.subject.rdnSequence”, “NEW”, 1);

SET OF: the same as SEQUENCE OF. Using “pkix.asn”:

result=asn1_write_value(cert, “tbsCertificate.subject.rdnSequence.?LAST”, “NEW”, 1);

RETURNS

ASN1_SUCCESS if the value was set, **ASN1_ELEMENT_NOT_FOUND** if name is not a valid element, and **ASN1_VALUE_NOT_VALID** if ivalue has a wrong format.

COPYRIGHT

SEE ALSO

The full documentation for **libtasn1** is maintained as a Texinfo manual. If the **info** and **libtasn1** programs are properly installed at your site, the command

info libtasn1

should give you access to the complete manual. As an alternative you may obtain the manual from:

http://www.gnu.org/software/libtasn1/manual/