

**NAME**

ASN1\_TIME\_set, ASN1\_TIME\_adj, ASN1\_TIME\_check, ASN1\_TIME\_set\_string, ASN1\_TIME\_print, ASN1\_TIME\_diff - ASN.1 Time functions.

**SYNOPSIS**

```
ASN1_TIME *ASN1_TIME_set(ASN1_TIME *s, time_t t);
ASN1_TIME *ASN1_TIME_adj(ASN1_TIME *s, time_t t,
int offset_day, long offset_sec);
int ASN1_TIME_set_string(ASN1_TIME *s, const char *str);
int ASN1_TIME_check(const ASN1_TIME *t);
int ASN1_TIME_print(BIO *b, const ASN1_TIME *s);

int ASN1_TIME_diff(int *pday, int *psec,
const ASN1_TIME *from, const ASN1_TIME *to);
```

**DESCRIPTION**

The function *ASN1\_TIME\_set()* sets the ASN1\_TIME structure *s* to the time represented by the time\_t value *t*. If *s* is NULL a new ASN1\_TIME structure is allocated and returned.

*ASN1\_TIME\_adj()* sets the ASN1\_TIME structure *s* to the time represented by the time **offset\_day** and **offset\_sec** after the time\_t value *t*. The values of **offset\_day** or **offset\_sec** can be negative to set a time before *t*. The **offset\_sec** value can also exceed the number of seconds in a day. If *s* is NULL a new ASN1\_TIME structure is allocated and returned.

*ASN1\_TIME\_set\_string()* sets ASN1\_TIME structure *s* to the time represented by string *str* which must be in appropriate ASN.1 time format (for example YYMMDDHHMMSSZ or YYYYMMDDHHMMSSZ).

*ASN1\_TIME\_check()* checks the syntax of ASN1\_TIME structure *s*.

*ASN1\_TIME\_print()* prints out the time *s* to BIO *b* in human readable format. It will be of the format MMM DD HH:MM:SS YYYY [GMT], for example “Feb 3 00:55:52 2015 GMT” it does not include a newline. If the time structure has invalid format it prints out “Bad time value” and returns an error.

*ASN1\_TIME\_diff()* sets **\*pday** and **\*psec** to the time difference between **from** and **to**. If **to** represents a time later than **from** then one or both (depending on the time difference) of **\*pday** and **\*psec** will be positive. If **to** represents a time earlier than **from** then one or both of **\*pday** and **\*psec** will be negative. If **to** and **from** represent the same time then **\*pday** and **\*psec** will both be zero. If both **\*pday** and **\*psec** are non-zero they will always have the same sign. The value of **\*psec** will always be less than the number of seconds in a day. If **from** or **to** is NULL the current time is used.

**NOTES**

The ASN1\_TIME structure corresponds to the ASN.1 structure **Time** defined in RFC5280 et al. The time setting functions obey the rules outlined in RFC5280: if the date can be represented by UTCTime it is used, else GeneralizedTime is used.

The ASN1\_TIME structure is represented as an ASN1\_STRING internally and can be freed up using *ASN1\_STRING\_free()*.

The ASN1\_TIME structure can represent years from 0000 to 9999 but no attempt is made to correct ancient calendar changes (for example from Julian to Gregorian calendars).

Some applications add offset times directly to a time\_t value and pass the results to *ASN1\_TIME\_set()* (or equivalent). This can cause problems as the time\_t value can overflow on some systems resulting in unexpected results. New applications should use *ASN1\_TIME\_adj()* instead and pass the offset value in the **offset\_sec** and **offset\_day** parameters instead of directly manipulating a time\_t value.

**BUGS**

*ASN1\_TIME\_print()* currently does not print out the time zone: it either prints out “GMT” or nothing. But all certificates complying with RFC5280 et al use GMT anyway.

**EXAMPLES**

Set a time structure to one hour after the current time and print it out:

```
#include <time.h>
#include <openssl/asn1.h>
ASN1_TIME *tm;
time_t t;
BIO *b;
t = time(NULL);
tm = ASN1_TIME_adj(NULL, t, 0, 60 * 60);
b = BIO_new_fp(stdout, BIO_NOCLOSE);
ASN1_TIME_print(b, tm);
ASN1_STRING_free(tm);
BIO_free(b);
```

Determine if one time is later or sooner than the current time:

```
int day, sec;

if (!ASN1_TIME_diff(&day, &sec, NULL, to))
/* Invalid time format */

if (day > 0 || sec > 0)
printf("Later\n");
else if (day < 0 || sec < 0)
printf("Sooner\n");
else
printf("Same\n");
```

**RETURN VALUES**

*ASN1\_TIME\_set()* and *ASN1\_TIME\_adj()* return a pointer to an *ASN1\_TIME* structure or *NULL* if an error occurred.

*ASN1\_TIME\_set\_string()* returns 1 if the time value is successfully set and 0 otherwise.

*ASN1\_TIME\_check()* returns 1 if the structure is syntactically correct and 0 otherwise.

*ASN1\_TIME\_print()* returns 1 if the time is successfully printed out and 0 if an error occurred (I/O error or invalid time format).

*ASN1\_TIME\_diff()* returns 1 for success and 0 for failure. It can fail if the pass *ASN1\_TIME* structure has invalid syntax for example.