NAME

openssl_user_macros, OPENSSL_API_COMPAT, OPENSSL_NO_DEPRECATED - User defined macros

DESCRIPTION

User defined macros allow the programmer to control certain aspects of what is exposed by the OpenSSL headers.

NOTE: to be effective, a user defined macro *must be defined before including any header file that depends on it*, either in the compilation command ("cc -DMACRO=value") or by defining the macro in source before including any headers.

Other manual pages may refer to this page when declarations depend on user defined macros.

The macros

OPENSSL_API_COMPAT

The value is a version number, given in one of the following two forms:

"0xMNNFF000L"

This is the form supported for all versions up to 1.1.x, where "M" represents the major number, "NN" represents the minor number, and "FF" represents the fix number, as a hexadecimal number. For version 1.1.0, that’s "0x10100000L".

Any version number may be given, but these numbers are the current known major deprecation points, making them the most meaningful:

"0x00908000L" (version 0.9.8)
"0x10000000L" (version 1.0.0)
"0x10100000L" (version 1.1.0)

For convenience, higher numbers are accepted as well, as long as feasible. For example, "0x60000000L" will work as expected. However, it is recommended to start using the second form instead:

"mmnnpp"

This form is a simple decimal number calculated with this formula:

major * 10000 + *minor* * 100 + *patch*

where *major*, *minor* and *patch* are the desired major, minor and patch components of the

version number. For example:

30000 corresponds to version 3.0.0
10002 corresponds to version 1.0.2
420101 corresponds to version 42.1.1

If **OPENSSL_API_COMPAT** is undefined, this default value is used in its place: 30000

OPENSSL_NO_DEPRECATED

If this macro is defined, all deprecated public symbols in all OpenSSL versions up to and including the version given by **OPENSSL_API_COMPAT** (or the default value given above, when **OPENSSL_API_COMPAT** isn't defined) will be hidden.

COPYRIGHT