



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 'BN_CTX_secure_new.3ossl'

\$ man BN_CTX_secure_new.3ossl

BN_CTX_NEW(3ossl) OpenSSL BN_CTX_NEW(3ossl)

NAME

BN_CTX_new_ex, BN_CTX_new, BN_CTX_secure_new_ex, BN_CTX_secure_new,
BN_CTX_free - allocate and free BN_CTX structures

SYNOPSIS

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

```
BN_CTX *BN_CTX_new_ex(OSSL_LIB_CTX *ctx);
```

```
BN_CTX *BN_CTX_new(void);
```

```
BN_CTX *BN_CTX_secure_new_ex(OSSL_LIB_CTX *ctx);
```

```
BN_CTX *BN_CTX_secure_new(void);
```

```
void BN_CTX_free(BN_CTX *c);
```

DESCRIPTION

A `BN_CTX` is a structure that holds `BIGNUM` temporary variables used by library functions. Since dynamic memory allocation to create `BIGNUMs` is rather expensive when used in conjunction with repeated subroutine calls, the `BN_CTX` structure is used.

`BN_CTX_new_ex()` allocates and initializes a `BN_CTX` structure for the given library context `ctx`. The `<ctx>` value may be `NULL` in which case the default library context will be used. `BN_CTX_new()` is the same as `BN_CTX_new_ex()` except that the default library context is always used.

`BN_CTX_secure_new_ex()` allocates and initializes a `BN_CTX` structure but uses the secure heap (see `CRYPTO_secure_malloc(3)`) to hold the `BIGNUMs` for the given library context `ctx`. The `<ctx>` value may be `NULL` in which case the default library context will be used. `BN_CTX_secure_new()` is the same as `BN_CTX_secure_new_ex()` except that the default library context is always used.

`BN_CTX_free()` frees the components of the `BN_CTX` and the structure itself. Since `BN_CTX_start()` is required in order to obtain `BIGNUMs` from the `BN_CTX`, in most cases `BN_CTX_end()` must be called before the `BN_CTX` may be freed by `BN_CTX_free()`. If `c` is `NULL`, nothing is done.

A given `BN_CTX` must only be used by a single thread of execution. No locking is performed, and the internal pool allocator will not properly handle multiple threads of execution.

RETURN VALUES

`BN_CTX_new()` and `BN_CTX_secure_new()` return a pointer to the `BN_CTX`. If the allocation fails, they return `NULL` and sets an error code that can be obtained by `ERR_get_error(3)`.

`BN_CTX_free()` has no return values.

REMOVED FUNCTIONALITY

```
void BN_CTX_init(BN_CTX *c);
```

BN_CTX_init() is no longer available as of OpenSSL 1.1.0. Applications should replace use of BN_CTX_init with BN_CTX_new instead:

```
BN_CTX *ctx;  
ctx = BN_CTX_new();  
if (!ctx)  
    /* error */  
...  
BN_CTX_free(ctx);
```

SEE ALSO

ERR_get_error(3), BN_add(3), BN_CTX_start(3)

HISTORY

BN_CTX_init() was removed in OpenSSL 1.1.0.

COPYRIGHT

Copyright 2000-2020 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>.