

**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 **BIGNUM**s 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 **BIGNUM**s 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 **BIGNUM**s 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>.