

Buffers

This page contains the primitives and their associated functions for buffer primitives within blib. While the measured buffers support the valloc family of allocators, the Measured Buffer helper functions all use the halloc family, so halloc should be used where possible.

MeasuredBuffers

A general-purpose container for a defined number of MeasuredBuffers.

```
typedef struct _Container_MeasuredBuffer {  
    DWORD count;  
    MeasuredBuffer** members;  
} Strings, MeasuredBuffers;
```

Initialisers

MeasuredBuffers* hallocMeasuredBuffers(DWORD count)

Allocates the measured buffers on the heap using the halloc family.

void hfreeMeasuredBuffers(MeasuredBuffers* buffers)

Frees the container and the subsequent pointers for each of the measured buffers allocated within the container IF the buffer is not NULL.

MeasuredBuffer

A general-purpose buffer with an assigned length up to `DWORD` bytes.

```
typedef struct _MeasuredBuffer {  
    DWORD length;  
    char* buffer;  
} MeasuredBuffer, String;
```

Initialisers

MeasuredBuffer* vallocMeasuredBuffer(DWORD bytes)

Allocated a measured buffer `DWORD` bytes using `valloc`.

void vfreeMeasuredBuffer(MeasuredBuffer* buffer)

Frees a measured buffer AND its content buffer using the valloc family.

MeasuredBuffer* hallocMeasuredBuffer(DWORD bytes)

Allocated a measured buffer `DWORD` bytes using `halloc`.

void hfreeMeasuredBuffer(MeasuredBuffer* buffer);

Frees a measured buffer AND its content buffer using the halloc family.

MeasuredBuffer*

bDeepCopyMeasuredBuffer(MeasuredBuffer* src)

Deep Copies a `Measured Buffer` 'src' and returns the pointer. This function uses the halloc family.

void bDeepCopyMeasuredBufferBuffer(MeasuredBuffer* dest, MeasuredBuffer* src);

Deep copies a `Measured Buffer` 'src' to the `Measured Buffer` 'dest'. This function allocates dest->buffer using the halloc family with the length of the original buffer. This frees the dest->buffer if it exists using halloc first.

Methods

BOOL bEncryptMeasuredBufferEx(void* algorithm, long* key, unsigned int keyLength, MeasuredBuffer* buffer)

Symmetrically encrypts or decrypts a `MeasuredBuffer` using the user supplied void `algorithm`.

DWORD bInterpetMeasuredBuffer(char* data, MeasuredBuffer* buffer)

Initialises and popualtes a measured buffer from data using `valloc`.

**DWORD bWriteMeasuredBuffer(HANDLE hFile,
MeasuredBuffer* buffer)**

Writes a measured buffer `buffer` to a file `hFile`.

**DWORD bReadMeasuredBuffer(HANDLE hFile,
MeasuredBuffer* buffer)**

Writes a measured buffer `buffer` to a file `hFile`.

CryptoBuffer

A 'cryptographic' primitive composed of two `MeasuredBuffer`s and a metadata element for an encryption type.

```
typedef enum _BLIB_ENCRYPTION_METHOD {  
    BLIB_ENCRYPTION_NONE,  
    BLIB_ENCRYPTION_UNDEFINED,  
    BLIB_ENCRYPTION_STATIC_XOR,  
} BLIB_ENCRYPTION_METHOD;  
  
typedef struct _EncryptedBuffer{  
    BLIB_ENCRYPTION_METHOD encryptionMethod;  
    MeasuredBuffer keyBuffer;  
    MeasuredBuffer dataBuffer;  
} CryptoBuffer;
```

Methods

DWORD bSizeOfCryptoBuffer(CryptoBuffer* buffer)

Returns the `DWORD` size of the entire CryptoBuffer struct including all members and their respective buffers.

DWORD bInterpretCryptoBuffer(char* data, CryptoBuffer* buffer)

Interprets the memory located in `data` and populates the fields into `buffer` using `bInterpretMeasuredBuffer`.

DWORD bWriteCryptoBuffer(HANDLE hFile, CryptoBuffer* buffer)

Writes a `CryptoBuffer` to a file.

DWORD bReadCryptoBuffer(HANDLE hFile, CryptoBuffer* buffer)

Reads a `CryptoBuffer` to a file.

BOOL bEncryptBuffer(CryptoBuffer* buffer)

Symmetrically encrypts or decrypts a `CryptoBuffer` using the accompanying method defined in the `encryptionMethod` field of the `CryptoBuffer`.

BOOL bEncryptBufferEx(void* algorithm, CryptoBuffer* buffer)

Symmetrically encrypts or decrypts a `CryptoBuffer` using the user supplied void `algorithm`.

Revision #8

Created 2 August 2024 18:20:50 by lepus

Updated 11 August 2024 08:46:41 by lepus