Package <default>

Class Diagram Summary	
<default>	

Class Summary	
bufferType	A buffer type has a pointer to an object of CGlobalMem and a flag that indiciates, if the object is valid (=used).
CGlobalMem	This class provides exactly one memory buffer which can be used as global memory within one (Windows) application.
GlobalMemory_dll_Export	This is a collection of functions that are exported by the dll (and not a class) and this is mainly a wrapper around the CGlobalMem class.
SharedMemory_dll_Export	This is a collection of functions that are exported by the dll (and not a class).

<default>

Class Diagram <default>

Class Summary	
bufferType	A buffer type has a pointer to an object of CGlobalMem and a flag that indiciates, if the object is valid (=used).
CGlobalMem	This class provides exactly one memory buffer which can be used as global memory within one (Windows) application.
GlobalMemory_dll_Export	This is a collection of functions that are exported by the dll (and not a class) and this is mainly a wrapper around the CGlobalMem class.
SharedMemory_dll_Export	This is a collection of functions that are exported by the dll (and not a class).

Class bufferType

---

class bufferType

A buffer type has a pointer to an object of CGlobalMem and a flag that indiciates, if the object is valid (=used).

Field Summary	
`CGlobalMem *`	**buffer** A pointer to a CGlobalMem object.
`char`	**used** 1: object is valid and in use 0: object does not exist

Field Detail

buffer

public CGlobalMem * **buffer**

A pointer to a CGlobalMem object.
label Provide a CGlobalMem object
supplierCardinality 0..1

---

used

public char **used**

1: object is valid and in use 0: object does not exist

Association Links

3, bufferType

to **Class** CGlobalMem

A pointer to a CGlobalMem object.

Label Provide a CGlobalMem object
Supplier Cardinality 0..1

Class CGlobalMem

---

class CGlobalMem

This class provides exactly one memory buffer which can be used as global memory within one (Windows) application. This class is used by the dll to manage global memory buffers.

Author:
Dietrich Beck
Version: 07−APR−2004

Field Summary	
`private unsigned char *`	**MemBuffer** Memory buffer
`private char [32+1]`	**MemName** Name of a global memory buffer.
`private unsigned long`	**MemSize** Size of global memory buffer

Constructor Summary
CGlobalMem() Provides a global memory buffer.

Destructor Summary
~CGlobalMem()

Method Summary	
`unsigned long`	**CreateMem** (unsigned long Size, char * Name) Creates a global memory buffer.
`unsigned long`	**GetMem**(unsigned long Offset, unsigned long Length, unsigned char * Buffer) Copies data from the global memory to a buffer Returns the number of bytes copied or 0, if not successful.
`void`	**GetMemName** (char * Name) Gets the name of the memory buffer.
`unsigned long`	**GetMemPointer** () Returns the pointer to the global memory buffer.
`unsigned long`	**GetMemSize** () Gets the size of the global memory buffer.
`unsigned long`	**InitMem**(unsigned char Value) Initializes the global memory with the specified value.
`unsigned long`	**SetMem** (unsigned long Offset, unsigned long Length, unsigned char * Buffer) Copies a buffer into the global memory buffer.

Field Detail

MemBuffer

private unsigned char ***MemBuffer**

Memory buffer

MemName

private char [32+1] **MemName**

Name of a global memory buffer. Note that the name is limited to 32 characters.

MemSize

private unsigned long**MemSize**

Size of global memory buffer

Constructor Detail

CGlobalMem

public **CGlobalMem**()

Provides a global memory buffer.

Method Detail

~CGlobalMem

public **~CGlobalMem**()

Method Detail

CreateMem

public unsigned long**CreateMem**(unsigned long Size, char * Name)

Creates a global memory buffer. The name of the memory buffer is an optional feature and limited to 32 characters.. If the global memory has not been created, it is created. If the global memory is already existing and the specified size equals the size of the global memory, no new memory is allocated but the pointer to the existing memory is returned. If the global memory is already existing but has a different size, 0 is returned. Returns the pointer to the global memory or 0, if not successful.

GetMem

public unsigned long**GetMem**(unsigned long Offset, unsigned long Length, unsigned char * Buffer)

Copies data from the global memory to a buffer Returns the number of bytes copied or 0, if not successful.

GetMemName

public void **GetMemName** (char * Name)

Gets the name of the memory buffer. The name has a length of up to 32 characters.

GetMemPointer

public unsigned long**GetMemPointer** ()

Returns the pointer to the global memory buffer. Returns 0, if not successful.

GetMemSize

public unsigned long**GetMemSize** ()

Gets the size of the global memory buffer.

InitMem

public unsigned long**InitMem**(unsigned char Value)

Initializes the global memory with the specified value. Returns the size of the global memory in bytes or 0, if not successful.

SetMem

public unsigned long**SetMem** (unsigned long Offset, unsigned long Length, unsigned char * Buffer)

> Copies a buffer into the global memory buffer. Returns the number of bytes copied or 0, if not successful.

Class GlobalMemory_dll_Export

---

class GlobalMemory_dll_Export

This is a collection of functions that are exported by the dll (and not a class) and this is mainly a wrapper around the CGlobalMem class. The exported dll supports many memory buffers, each buffer is an object of the CGlobalMem class. The number of buffers is limited (presently 131072). Important: Note that the global memory is only shared within one application and NOT(!) between different Windows applications.

Author:
Dietrich Beck
Version: 07−APR−2004

Field Summary	
`private bufferType *`	**lnkbufferType** The exported dll functions uses an array of the buffer type to manage a lot of buffers.

Method Summary	
`unsigned long`	**CreateBuffer** (long * ID, unsigned long Size, char * Name) Creates a global memory buffer.
`long`	**DestroyBuffer** (long ID) Destroys a global memory buffer.
`unsigned long`	**GetBuffer** (long ID, unsigned long Offset, unsigned long Length, unsigned char * Buffer) Copies data from a global memory buffer to a buffer.
`long`	**GetBufferID** (char * Name) Gets the ID of a global memory buffer.

Method Summary	
`long`	**GetBufferName** (long ID, char * Name) Gets the name of a global memory buffer.
`unsigned long`	**GetBufferPointer** (long ID) Gets the pointer to the global memory buffer.
`unsigned long`	**GetBufferSize** (long ID) Gets the size of a global memory buffer.
`int`	**GetProcCounter** (void) Gets the number of processes that are connected to the dll.
`unsigned long`	**InitBuffer** (long ID, unsigned char Value) Initializes a global memory buffer with the specified value.
`unsigned long`	**SetBuffer** (long ID, unsigned long Offset, unsigned long Length, unsigned char * Buffer) Copies a buffer into a global memory buffer.

Field Detail

lnkbufferType

private bufferType * **lnkbufferType**

The exported dll functions uses an array of the buffer type to manage a lot of buffers. The number of buffers (not their size) is limited (more than 10E5).
label Use array of bufferType
directed
supplierCardinality *

Method Detail

CreateBuffer

public unsigned long **CreateBuffer** (long * ID, unsigned long Size, char * Name)

Creates a global memory buffer. The name of the memory buffer is an optional feature and limited to 32 characters. If the name is specified, another buffer with the same name must not exist. This function always creates a new buffer. In order to link to an existing buffer one can use the functions GetBuffID, GetBufferName and GetBufferPointer. Returns the pointer to the global memory or 0, if not successful.

DestroyBuffer

public long **DestroyBuffer** (long ID)

Destroys a global memory buffer. Returns the ID of the buffer or −1, if not successful.

GetBuffer

public unsigned long **GetBuffer** (long ID, unsigned long Offset, unsigned long Length, unsigned char * Buffer)

Copies data from a global memory buffer to a buffer. Returns the number of bytes copied or 0, if not successful.

GetBufferID

public long **GetBufferID** (char * Name)

Gets the ID of a global memory buffer. Returns the ID of the buffer or −1, if not successful.

GetBufferName

public long **GetBufferName** (long ID, char * Name)

Gets the name of a global memory buffer. The name has a length of up to 32 characters. Returns the ID of the buffer or −1, if not successful.

GetBufferPointer

public unsigned long**GetBufferPointer** (long ID)

> Gets the pointer to the global memory buffer. Returns the pointer to the buffer or 0, if not successful.

GetBufferSize

public unsigned long**GetBufferSize** (long ID)

> Gets the size of a global memory buffer. Gets the size of the buffer or 0, if not successful.

GetProcCounter

public int **GetProcCounter**(void)

> Gets the number of processes that are connected to the dll.

InitBuffer

public unsigned long**InitBuffer**(long ID, unsigned char Value)

> Initializes a global memory buffer with the specified value. Returns the size of the global memory buffer in bytes or 0, if not successful.

SetBuffer

public unsigned long**SetBuffer** (long ID, unsigned long Offset, unsigned long Length, unsigned char * Buffer)

> Copies a buffer into a global memory buffer. Returns the number of bytes copied or 0, if not successful.

Association Links

to **Class** bufferType

The exported dll functions uses an array of the buffer type to manage a lot of buffers. The number of buffers (not their size) is limited (more than 10E5).

Label Use array of bufferType
Supplier Cardinality *
Is directed

Class SharedMemory_dll_Export

---

class SharedMemory_dll_Export

This is a collection of functions that are exported by the dll (and not a class). By this, the dll provides memory with a fixed size (presently 1048576, check using the function GetSharedBufferSize) which can be shared between different Windows applications.
Important: Sharing memory between different applications is dangerous and should be restricted to special cases.

Method Summary	
`int`	**GetProcCounter**(void) Gets the number of processes that are connected to the dll.
`unsigned long`	**GetSharedBuffer** (unsigned long Offset, unsigned long Length, unsigned char * Buffer) Copies data from the shared memory to a buffer.
`unsigned long`	**GetSharedBufferPointer** () Gets the pointer to the shared memory.
`unsigned long`	**GetSharedBufferSize** () Gets the size of the shared memory.
`unsigned long`	**SetSharedBuffer** (unsigned long Offset, unsigned long Length, unsigned char * Buffer) Copies a buffer into a the shared memory.

Method Detail

GetProcCounter

public int **GetProcCounter**(void)

Gets the number of processes that are connected to the dll.

GetSharedBuffer

public unsigned long**GetSharedBuffer** (unsigned long Offset, unsigned long Length, unsigned char * Buffer)

Copies data from the shared memory to a buffer. Returns the number of bytes copied or 0, if not successful.

GetSharedBufferPointer

public unsigned long**GetSharedBufferPointer** ()

Gets the pointer to the shared memory. Returns the pointer to the buffer or 0, if not successful.

GetSharedBufferSize

public unsigned long**GetSharedBufferSize** ()

Gets the size of the shared memory. Gets the size of the buffer or 0, if not successful.

SetSharedBuffer

public unsigned long**SetSharedBuffer** (unsigned long Offset, unsigned long Length, unsigned char * Buffer)

Copies a buffer into a the shared memory. Returns the number of bytes copied or 0, if not successful.