

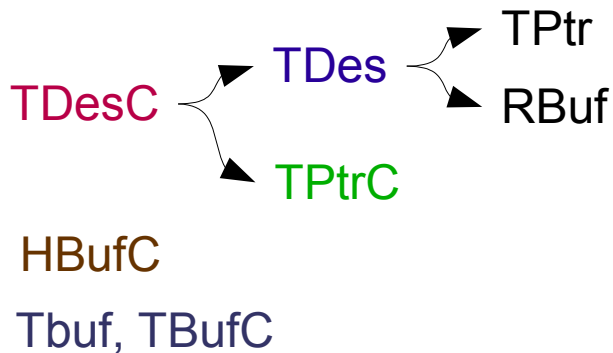
Descriptors

Descriptor - an integer, string or other small data value - which refers to one of several objects allocated to a program.

Ex: A reference to an area of memory.

Ex: Small Object that either contains or refers to an array in memory.

Class Tree



TDesC - base. abstract. Mem-**layout-0**.

TDes - abstract. Mem-**layout-2**. Modifies data.

TPtr - pointer descriptor. Mem-**layout-3,4**. Modifies data.

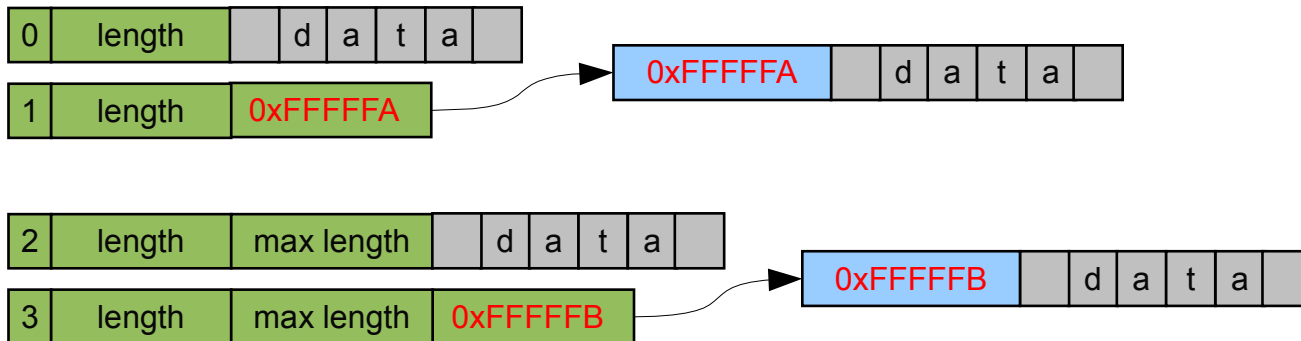
RBuf - resizable pointer descriptor. Mem-**layout-3**. Modifies data.

TPtrC - pointer descriptor. Mem-**layout-1**.

HBufC - buffer allocated in heap. max-length gets by checking allocated cels in memory. Modifies data. Mem-**layout-1**.

TBuf - buffer with fixed length. Mem-**layout-2**.

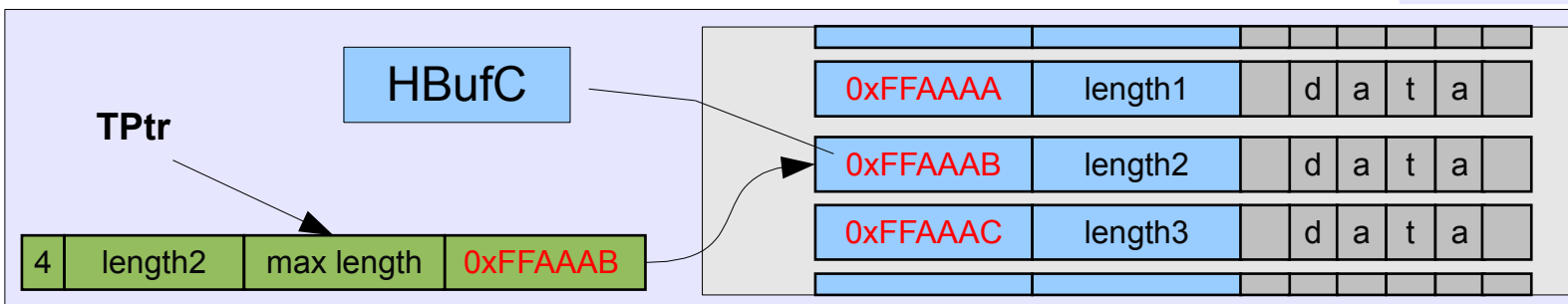
Memory Layout



How to pick:

- construct descriptor, use and forget it
- no bothers with memory allocations
- TBuf<maxLength> buffer;**

- have data that is not descriptor but needs to be passed to a function that expects descriptor
- get a substring of a descriptor
- TPtrC**



- want some raw memory to be used as the content of a descriptor
- TPtr**

Descriptor Usage

TDesC – being the base for all the descriptors, is used to pass any descriptors as parameter to function that wants a descriptor.

How to use:

```
const TDesC& getConstData() {}

void passInConstParameter( const TDesC& parameter ) {}

void passInModifiableParameter( TDes& parameter ) {}

TDes& getBufferToWriteInto() {}

RBuf& getBufferToWriteIntoAndResizeIt() {}

HBufC& allocateDescriptor() {}

void allocateDescriptor( RBuf& parameter ) {}

class SuperDuperClass
{
    public:
        // this will hold integers of max 4 digits
        TBuf<4> _mySmallBuffer;

        // this will hold integers of full range
        TBuf<sizeof(TInt)> _fullBlownBuffer;

    public:
        void doSomethingWithinThisClass()
        {
            // this is a temporary buffer with lifetime of the method call
            TBuf<5> localTemporaryBuffer;
            localTemporaryBuffer = _mySmallBuffer;
        }
};
```