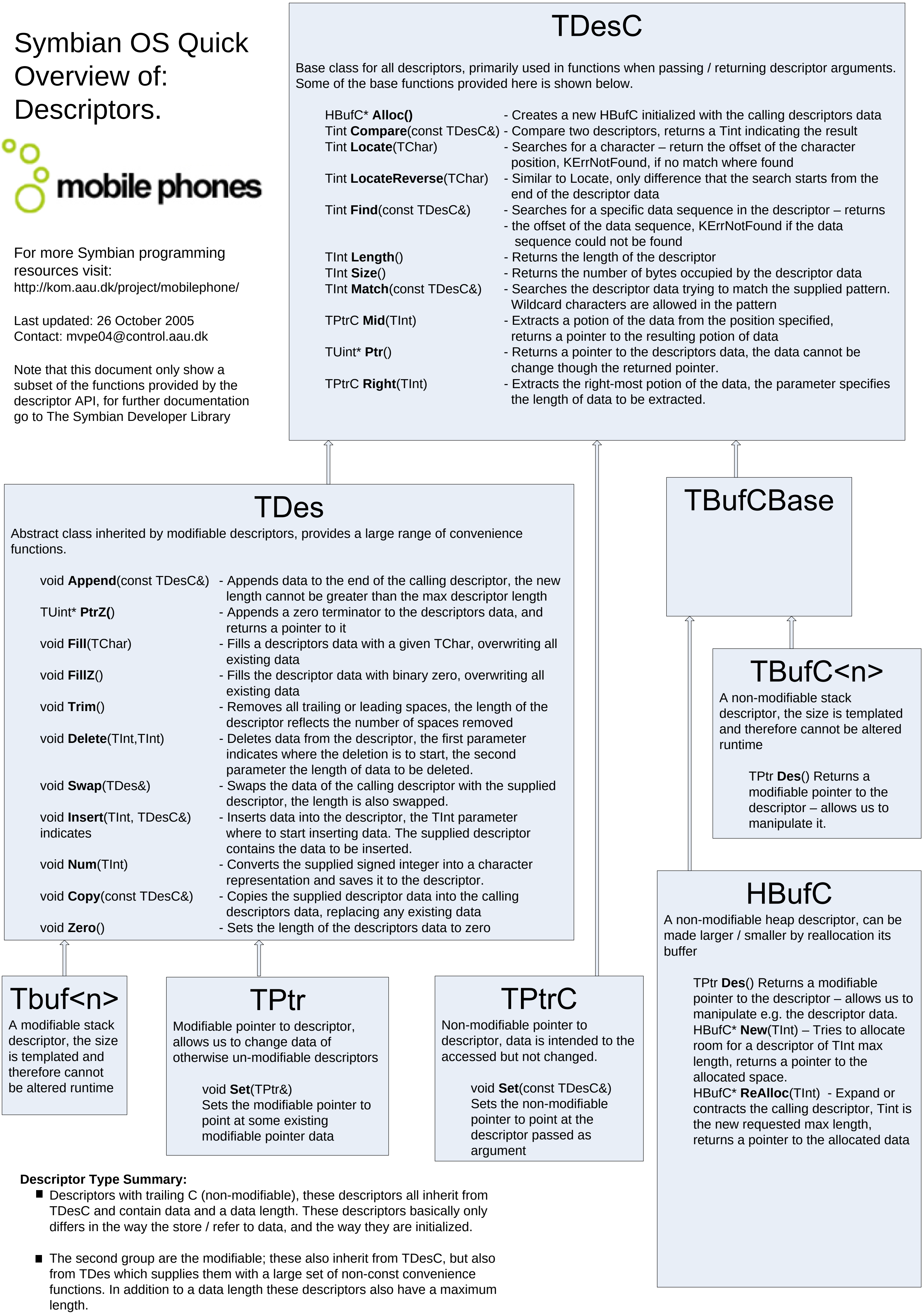
Symbian OS Quick Overview of: Descriptors.

mobile phones

For more Symbian programming resources visit:
http://kom.aau.dk/project/mobilephone/

Last updated: 26 October 2005
Contact: mvpe04@control.aau.dk

Note that this document only show a subset of the functions provided by the descriptor API, for further documentation go to The Symbian Developer Library



TDesC

Base class for all descriptors, primarily used in functions when passing / returning descriptor arguments. Some of the base functions provided here is shown below.

HBufC* **Alloc()**	- Creates a new HBufC initialized with the calling descriptors data
Tint **Compare**(const TDesC&)	- Compare two descriptors, returns a Tint indicating the result
Tint **Locate**(TChar)	- Searches for a character – return the offset of the character position, KErrNotFound, if no match where found
Tint **LocateReverse**(TChar)	- Similar to Locate, only difference that the search starts from the end of the descriptor data
Tint **Find**(const TDesC&)	- Searches for a specific data sequence in the descriptor – returns - the offset of the data sequence, KErrNotFound if the data sequence could not be found
TInt **Length**()	- Returns the length of the descriptor
TInt **Size**()	- Returns the number of bytes occupied by the descriptor data
TInt **Match**(const TDesC&)	- Searches the descriptor data trying to match the supplied pattern. Wildcard characters are allowed in the pattern
TPtrC **Mid**(TInt)	- Extracts a potion of the data from the position specified, returns a pointer to the resulting potion of data
TUint* **Ptr**()	- Returns a pointer to the descriptors data, the data cannot be change though the returned pointer.
TPtrC **Right**(TInt)	- Extracts the right-most potion of the data, the parameter specifies the length of data to be extracted.

TDes

Abstract class inherited by modifiable descriptors, provides a large range of convenience functions.

void **Append**(const TDesC&)	- Appends data to the end of the calling descriptor, the new length cannot be greater than the max descriptor length
TUint* **PtrZ()**	- Appends a zero terminator to the descriptors data, and returns a pointer to it
void **Fill**(TChar)	- Fills a descriptors data with a given TChar, overwriting all existing data
void **FillZ**()	- Fills the descriptor data with binary zero, overwriting all existing data
void **Trim**()	- Removes all trailing or leading spaces, the length of the descriptor reflects the number of spaces removed
void **Delete**(TInt,TInt)	- Deletes data from the descriptor, the first parameter indicates where the deletion is to start, the second parameter the length of data to be deleted.
void **Swap**(TDes&)	- Swaps the data of the calling descriptor with the supplied descriptor, the length is also swapped.
void **Insert**(TInt, TDesC&) indicates	- Inserts data into the descriptor, the TInt parameter where to start inserting data. The supplied descriptor contains the data to be inserted.
void **Num**(TInt)	- Converts the supplied signed integer into a character representation and saves it to the descriptor.
void **Copy**(const TDesC&)	- Copies the supplied descriptor data into the calling descriptors data, replacing any existing data
void **Zero**()	- Sets the length of the descriptors data to zero

TBufCBase

TBufC<n>

A non-modifiable stack descriptor, the size is templated and therefore cannot be altered runtime

TPtr **Des**() Returns a modifiable pointer to the descriptor – allows us to manipulate it.

Tbuf<n>

A modifiable stack descriptor, the size is templated and therefore cannot be altered runtime

TPtr

Modifiable pointer to descriptor, allows us to change data of otherwise un-modifiable descriptors

void **Set**(TPtr&)
Sets the modifiable pointer to point at some existing modifiable pointer data

TPtrC

Non-modifiable pointer to descriptor, data is intended to the accessed but not changed.

void **Set**(const TDesC&)
Sets the non-modifiable pointer to point at the descriptor passed as argument

HBufC

A non-modifiable heap descriptor, can be made larger / smaller by reallocation its buffer

TPtr **Des**() Returns a modifiable pointer to the descriptor – allows us to manipulate e.g. the descriptor data.
HBufC* **New**(TInt) – Tries to allocate room for a descriptor of TInt max length, returns a pointer to the allocated space.
HBufC* **ReAlloc**(TInt) - Expand or contracts the calling descriptor, Tint is the new requested max length, returns a pointer to the allocated data

Descriptor Type Summary:

- Descriptors with trailing C (non-modifiable), these descriptors all inherit from TDesC and contain data and a data length. These descriptors basically only differs in the way the store / refer to data, and the way they are initialized.
- The second group are the modifiable; these also inherit from TDesC, but also from TDes which supplies them with a large set of non-const convenience functions. In addition to a data length these descriptors also have a maximum length.