Basic Data Types

- **TIntX** and **TUintX** (for X = 8, 16 and 32) for 8-, 16- and 32-bit signed and unsigned integers respectively. Unless you have a good reason to do so, such as for size optimization or compatibility, you should use the non-specific TInt or TUint types, which correspond to signed and unsigned 32-bit integers, respectively.
- **TInt64**. Releases of Symbian OS prior to v8.0 had no built-in ARM support for 64-bit arithmetic, so the TInt64 class implemented a 64-bit integer as two 32-bit values. On Symbian OS v8.0, TInt64 and TUInt64 are typedef'd to long long and use the available native 64-bit support.
- **TReal32** and **TReal64** (and TReal, which equates to TReal64) for single- and double-precision floating point numbers, equivalent to float and double respectively.1 Operations on these are likely to be slower than upon integers so you should try to avoid using them unless necessary.
- **TTextX** (for X = 8 or 16), for narrow and wide characters, correspond to 8-bit and 16-bit unsigned integers, respectively.
- **TAny*** should be used in preference to void*, effectively replacing it with a typedef'd "pointer to anything". TAny is thus equivalent to void but, in the context where void means "nothing", it is not necessary to replace the native void type. Thus, a function taking a void* pointer (to anything) and returning void (nothing) will typically have a signature as follows on Symbian OS:

  ```
  void TypicalFunction(TAny* aPointerParameter);
  ```

  This is the one exception to the rule of replacing a native type with a Symbian OS typedef; it occurs because void is effectively compiler-independent when referring to "nothing".
- **TBool** should be used for boolean types. For historical reasons TBool is equivalent to int and you should use the Symbian OS typedef'd values of ETrue (=1) and EFalse (=0). Bear in mind that C++ will interpret any nonzero value as true. For this reason, you should refrain from making direct comparisons with ETrue.
- Each TBool represents 32 bits, which may be quite wasteful of memory in classes with a number of flags representing state or settings.

(C) Copyright Nokia 2013. All rights reserved.

(C) Copyright Nokia 2013. All rights reserved.