



This article *does not* meet Forum Nokia Wiki quality standards or the information included may not be complete.

Please help improve this article by adding links and further information.

--Himsymbian 09:12, 21 February 2009 (EET)[**Fundamental_types_in_symbian**] 1] symbian os provides a set of typedef's for the built-in types which are defined in e32def.h for example : typedef signed char TInt8

2] there are different variants (alternatives or options) for each of the data types.

[INTEGER TYPES] - TInt8,TInt16,TInt32

3] in integer data types we are having :

1) TInt8,TInt16,TInt32 (signed integer types)

2) TUInt8,TUInt16,TUInt32 (unsigned integer types)

they have been typedef' d as follows :

a) typedef signed char TInt8 - [signed type]

TInt8 i;

Size of variable i : 8 bits (1 byte)

range of variable i : [-128 to +127]

typedef unsigned char TUInt8 - [unsigned type]

TUInt8 i;

Size of variable i : 8 bits (1 byte)

range of variable i : [0 to 255]

b) typedef short int TInt16

TInt16 i;

Size of variable i : 16 bits (2 bytes)

range of variable i : [-32768 to +32767]

typedef unsigned short int TUInt16

TUInt16 i;

Size of variable i : 16 bits (2 bytes)

range of variable i : [0 to +65535]

Fundamental_Types_in_symbian

c) typedef long int TInt32

TInt32 i;

Size of variable i : 32 bits (4 bytes)

range of variable i : [-2^{31} to $2^{31} - 1$] OR [-2147483648 to 2147483647]

typedef unsigned long int TUint32

TUint32 i;

Size of variable i : 32 bits (4 bytes)

range of variable i : [0 to $2^{32} - 1$] OR [0 to 4294967295]

These are termed as the specific variants. they are independent of the underlying implementation.

4] there are also generic variants of these data types ex: [TInt and TUint]

Size of these generic variants is the size of the [natural machine word length] on that particular system, which is atleast 32 bits (4 bytes).

They are dependant on the underlying implementation.

They have been typedef'ed as follows:

typedef signed int TInt

typedef unsigned int TUint

[INTEGER TYPES] - TInt64

5] a) Before symbian os version 8.0, there was no default ARM support for 64 bit arithmetic.

b) that is, operations on 64 bit values cannot be performed directly as with 32 bit, 16 bit or 8 bit values respectively.

c) so , 64 bit values were implemented as an object of class TInt64, where each object internally contains 2 unsigned 32 bit integers as follows:

TUint iLow; TUint iHigh;

6] a) After symbian os version 8.0, there is built in support for 64 bit arithmetic.

b) so 64 bit values were typedef'ed, instead of an object implementation containing 2 unsigned integers.

c] [64 bit values typedef as long]

typedef long long Int64

Fundamental_Types_in_symbian

```
typedef Int64 TInt64
```

```
typedef unsigned long long UInt64
```

```
typedef UInt64 TUInt64
```

[REAL TYPES] - TReal32,TReal64

7] a) Floating point numbers can be represented in symbian os using type TReal,TReal32 and TReal64.

b) they are typedef as follows:

```
typedef float TReal32
```

```
typedef double TReal64
```

c) by default, TReal is typedef as double.

```
typedef double TReal
```

d) symbian os does not have a dedicated floating point unit, so floating point calculations in symbian os are much slower as compared to integer calculations.

[CHARACTER TYPES] - TText,TText8 and TText16.

8] a) Characters in symbian are represented using type TText,TText8 and TText16.

they are typedef as follows:

b) typedef unsigned char TText8 where the character size is 8 bit (1 byte). It represents a narrow character.

TText8 has been typedef'd as unsigned char whose range is 0 to 255.

```
TText8 ch;
```

A variable of type TText8 can represent any character whose ascii number is between 0 to 255.

c) typedef unsigned short int TText16 where the character size is 16 bit (2 bytes). It represents a wide character.

TText16 has been typedef'd as unsigned short int whose range is 0 to 65535.

```
TText16 ch;
```

A variable of type TText16 can represent any unicode character whose unicode number is between 0 to 65535.

d) TText can also be used to represent a character.

Implicitly TText is mapped to TText8 for non-unicode builds, and for unicode builds

Fundamental_Types_in_symbian

TText is mapped to Text16.

[VOID TYPE] - TAny

9] a) In standard C++, void means nothing. that is when ever a function takes no argument or returns nothing it can be declared as follows:

```
void add(void);
```

b) In standard C++, (void *) means ?pointer to anything?, i.e pointer to any data type. It is also termed as a generic pointer.

c) In symbian C++ void has been typedef ?d as follows: typedef void TAny in e32def.h

d) In symbian C++, in context where void means ?nothing?,that is when a function takes no argument or it returns no argument as: void add(void);

[void] is used instead of [TAny] as void is effectively compiler independent when referring to ??nothing??.

e) In symbian C++, in context where a function takes void pointer as an argument (void *), that is ?pointer to anything? , or anywhere in the code, when we are using ?pointer to any thing?, [TAny *] is used instead of [void *] in symbian.

[BOOLEAN TYPE] - TBool

10] a) boolean values in symbian are represented using type TBool.

b) It has been typedef? d as follows: typedef int TBool in e32def.h

c) A variable of type TBool takes 32 bits, as it has been typedef to an int.

d) A variable of type TBool can have either of the two values [ETrue] or [EFalse]. ex : TBool var

a) var = ETrue

here the variable var takes 32 bits in size, and stores value 1 in those 32 bits. [ETrue is equivalent to 1]

b) var = EFalse

here the variable var takes 32 bits in size, and stores value 0 in those 32 bits. [EFalse is equivalent to 0]

11] [Comparison or relation between symbian os fundamental types and native C++ built types] [native types-symbian types]

1] signed char - TInt8

2] unsigned char - TUint8

3] short int - TInt16

4] unsigned short int - TUint16

Fundamental_Types_in_symbian

5] long int - TInt32

6] unsigned long int - TUint32

7] a) long long - Int64 b) Int64 - TInt64

8] a) unsigned long long - UInt64 b) UInt64 - TUint64

9] float - TReal32

10] double - TReal64

11] unsigned char - TText8

12] unsigned short int - TText16

13] void* - TAny*

14] int - TBool

12] You can mix symbian [C++ fundamental types] with [native C++ types], but it is more recommended as symbian os fundamental types as independant of the underlying implementation. --[Himsymbian](#) 09:12, 21 February 2009 (EET)

<http://himanshukar.wordpress.com/>
