
Subject: signature of the stou function

Posted by [hojtsy](#) on Fri, 03 Feb 2006 19:05:14 GMT

[View Forum Message](#) <> [Reply to Message](#)

Wouldn't it be better & more type-safe if the type of the endptr parameter in these functions would be const char **, and const wchar ** ?

```
unsigned stou(const char *s, void *endptr, unsigned base)
uint64 stou64(const char *s, void *endptr, unsigned base)
unsigned stou(const wchar *s, void *endptr, unsigned base)
```

Subject: Re: signature of the stou function

Posted by [rylek](#) on Sun, 05 Feb 2006 15:50:29 GMT

[View Forum Message](#) <> [Reply to Message](#)

Perhaps you're right. The main practical reason for the above declarations is to avoid the need to have two versions of the function with the 'endptr' argument of type char ** and const char ** (note that, unlike with char * and const char *, char ** cannot be automatically converted to const char **). But perhaps it's better to sacrifice three more lines of code for the sake of clarity, so I'll discuss it with Mirek and I guess we'll agree on changing it.

Regards

Tomas

Subject: Re: signature of the stou function

Posted by [hojtsy](#) on Mon, 06 Feb 2006 13:09:09 GMT

[View Forum Message](#) <> [Reply to Message](#)

Another thing I am finding strange is the completely different naming for string->unsigned int and string->signed int conversion functions. Different function name, different parameter name (radix/base). I believe they should rather follow the same naming scheme: ScanInt and ScanUInt.

```
unsigned    stou(const char *ptr, void *endptr = NULL, unsigned base = 10);
unsigned    stou(const byte *ptr, void *endptr = NULL, unsigned base = 10);
unsigned    stou(const wchar *ptr, void *endptr = NULL, unsigned base = 10);
int         ScanInt(const char *ptr, const char **endptr = NULL, int radix = 10);
int         ScanInt(const wchar *ptr, const wchar **endptr = NULL, int radix = 10);
```

Subject: Re: signature of the stou function

Posted by [mirek](#) on Mon, 06 Feb 2006 13:15:22 GMT

[View Forum Message](#) <> [Reply to Message](#)

...different function...

There is an important difference - ScanInt etc... are designed to return Null in the case there is no number (U++ way). That is quite a difference from stou etc., which just supplement C library. Note that there is no Null defined for unsigned...

Mirek

Subject: Re: signature of the stou function
Posted by [hojtsy](#) on Mon, 06 Feb 2006 13:26:09 GMT
[View Forum Message](#) <> [Reply to Message](#)

stou returns 0xFFFFFFFF in case of error. Maybe that could be considered as Null for unsigned. Anyway I think that converting a string to unsigned int, signed int, and double are similar operations and should have similar interface for the client code. So if you can not or do not want to provide an invalid return value, then one more parameter should be added for success indication, or the client code should check the endptr to see if the conversion succeeded. It is not really intuitive for the client programmer to call stou and check for 0xFFFFFFFF for unsigned integers, and call ScanInt and check for Null for signed integers.

Subject: Re: signature of the stou function
Posted by [mirek](#) on Mon, 06 Feb 2006 13:34:26 GMT
[View Forum Message](#) <> [Reply to Message](#)

I see your point, your arguments are correct, however I am not going to define (dword)Null as 0xffffffff - if there are any relations between Null-able types, Null should be < than all other values...

Mirek
