Subject: string filtering bug
Posted by Zbych on Tue, 21 Jul 2009 12:33:04 GMT
View Forum Message <> Reply to Message

Hi,

String filtering functions for some reason treat all input data as bytes. This causes incorrect

```
WString Filter(const wchar *s, int (*filter)(int))
{
 WString result;
 while(*s) {
  int c = (*filter)((char)*s++);
//                          ^^^^^^^^^ bug, should be wchar
  if(c) result.Cat(c);
 }
 return result;
}
```

```
String Filter(const char *s, int (*filter)(int))
{
 String result;
 while(*s) {
  int c = (*filter)((byte)*s++);
//                          ^^^^^^^ problem when s is UTF-8
  if(c) result.Cat(c);
 }
 return result;
}
```

Subject: Re: string filtering bug
Posted by mirek on Tue, 21 Jul 2009 15:17:50 GMT
View Forum Message <> Reply to Message

Zbych wrote on Tue, 21 July 2009 08:33Hi,

String filtering functions for some reason treat all input data as bytes. This causes incorrect

```
WString Filter(const wchar *s, int (*filter)(int))
{
 WString result;
 while(*s) {
```

```
  int c = (*filter)((char)*s++);
//                          ^^^^^^^^^ bug, should be wchar
  if(c) result.Cat(c);
 }
 return result;
}
```

```
String Filter(const char *s, int (*filter)(int))
{
 String result;
 while(*s) {
  int c = (*filter)((byte)*s++);
//                          ^^^^^^^ problem when s is UTF-8
  if(c) result.Cat(c);
 }
 return result;
}
```

Thanks. First one fixed (hopefuly), second one I have to think through...

Mirek

Subject: Re: string filtering bug
Posted by Zbych on Tue, 21 Jul 2009 19:28:30 GMT
View Forum Message <> Reply to Message

luzr wrote on Tue, 21 July 2009 17:17Thanks. First one fixed (hopefuly), second one I have to think through...

Thanks.
I think that the same kind of bug is in [] operator in String - it returns n-th byte instead of n-th letter.
This example works fine:

```
 SetDefaultCharset(CHARSET_UTF8);
 String first_name = "John";
 String second_name = "Wayne";
 String login = first_name[0] + second_name;
 PromptOK(login);
```

But this one doesn't:

```
 SetDefaultCharset(CHARSET_UTF8);
```

```
String second_name = "Wayne";
String login = first_name[0] + second_name;
PromptOK(login);
```

Subject: Re: string filtering bug
Posted by cbpporter on Tue, 21 Jul 2009 19:44:45 GMT
View Forum Message <> Reply to Message

This is not a bug, rather a design choice. String is not Utf8, it is 8 bit. You can store Utf8 in it, but U++ doesn't handle code points if you don't convert to WString, and even then only 16 bit, not true Unicode and will fail if you do real internationalization.

Generally you can store Utf8 without problems, but if you need to iterate over an Utf8 string you are left to your own devices. Utf8 (and even Utf32) are not indexable. There is no way to implement a fast []. There is a way to implement an amortized cost [], but the best way is to use an iterator which has excellent performance with the limitation that you can only go ahead or backwards in a linear fashion. Fortunately, in most cases this is what you need.

Subject: Re: string filtering bug
Posted by Zbych on Wed, 22 Jul 2009 08:34:33 GMT
View Forum Message <> Reply to Message

cbpporter wrote on Tue, 21 July 2009 21:44This is not a bug, rather a design choice.

Ok, I checked that there is information about this in help:
Quote:String works with 8 bit characters.

but I think that there should be a warning about String and UTF-8 (since UTF-8 is default encoding for UPP).

Subject: Re: string filtering bug
Posted by mirek on Sun, 26 Jul 2009 01:27:07 GMT
View Forum Message <> Reply to Message

Zbych wrote on Wed, 22 July 2009 04:34
but I think that there should be a warning about String and UTF-8 (since UTF-8 is default encoding for UPP).

OK, I have tried my best...