Subject: [SOLVED] String.GetCount with umlaut
Posted by forlano on Tue, 06 Sep 2011 21:12:14 GMT

View Forum Message <> Reply to Message

Hello,

I see only 6 characters. Where is the trick?

I am having problem with string aligning of text file in case of accented characters. If even GetCount give uncorrect answer I will not be able to correct the row lenght to be displayed adding additional space (it seems that for each accented characters a space is eated). By the way this was an old issue that I was never able to resolve in my application.


```
//SetDefaultCharset(CHARSET_UTF8);

ss << t.GetCount();
SaveFile("out.txt", ss);
```


Thanks,
Luigi

Subject: Re: String.GetCount with umlaut
Posted by forlano on Wed, 07 Sep 2011 06:20:24 GMT

View Forum Message <> Reply to Message

forlano wrote on Tue, 06 September 2011 23:12Hello,

I see only 6 characters. Where is the trick?

I am having problem with string aligning of text file in case of accented characters. If even GetCount give uncorrect answer I will not be able to correct the row lenght to be displayed adding additional space (it seems that for each accented characters a space is eated). By the way this was an old issue that I was never able to resolve in my application.

Here is what I mean with aligning characters

I tried to accomodate the names with %-32.32 but the accented characters are counted twice (!) so the visual effect was that of eaten space or shift to left.

Perhaps I must convert the name to some other encoding before to save them   let's go to try ... Anyway the number of chars I think should be calculated correctly under every encoding.

Luigi

File Attachments
1) 2011-09-07_081401.png, downloaded 978 times

Subject: Re: String.GetCount with umlaut
Posted by forlano on Wed, 07 Sep 2011 06:27:29 GMT
View Forum Message <> Reply to Message

forlano wrote on Wed, 07 September 2011 08:20
Perhaps I must convert the name to some other encoding before to save them   let go to try ...

Solved!

out << NFormat(" %-32.32s ", ToCharset(CHARSET_WIN1252, player[i].name, CHARSET_UTF8));

work
With notepad and wordpad the accent are ok and name aligned (problem solved after 5 years), but within my app (UTF8 encoded) they are disappeared. This last behaviour should be normal.

The new questions are now:
1) which CHARSET_WIN??? should I use for my text file in Windows in case of latin letter to accomodate the maximum number of accents (German, Italian, Danish, French...)?
2) do I need to convert even under Linux to prevent this problem?
I have no experience on this matter.
Luigi

Subject: Re: String.GetCount with umlaut
Posted by mirek on Wed, 07 Sep 2011 07:46:48 GMT
View Forum Message <> Reply to Message

forlano wrote on Tue, 06 September 2011 17:12Hello,

I see only 6 characters. Where is the trick?

GetCount or GetLength returns a number of _bytes_ in string.

When it is UTF-8 encoded, this is more than number of characters it represents.

Good way to convert is

.ToWString().GetCount()

or you can use "utf8len" function too..

(Anyway, thinking about it, perhaps I shall add "GetCharCount()" method to String...).

Mirek

Subject: Re: String.GetCount with umlaut
Posted by mr_ped on Wed, 07 Sep 2011 07:47:57 GMT
View Forum Message <> Reply to Message

UTF8 is multibyte encoding with variable length for different characters.

IMHO the most correct solution for you is to convert from UTF8 to WString (UCS2), which should be well enough to cover *all* latin-like alphabets, Russian's azbuka as well and some more. (I'm afraid UCS2 does not cover all Chinese characters/etc.. no time to google). Then work with WString to count characters, etc..

edit: Mirek beat me...

Quote:(Anyway, thinking about it, perhaps I shall add "GetCharCount()" method to String...).

Indeed for sure, actually UTF8 would love to have full set of functions like len/right/left/mid/etc.. whenever number of characters (position) is used as parameter. Plus special section in documentation to explain it.

Subject: Re: String.GetCount with umlaut
Posted by mirek on Wed, 07 Sep 2011 07:57:34 GMT
View Forum Message <> Reply to Message

mirek wrote on Wed, 07 September 2011 03:46
(Anyway, thinking about it, perhaps I shall add "GetCharCount()" method to String...).

Mirek

OK, you can now call GetCharCount() for String

Mirek

Subject: Re: String.GetCount with umlaut
Posted by forlano on Wed, 07 Sep 2011 08:09:11 GMT
View Forum Message <> Reply to Message

mirek wrote on Wed, 07 September 2011 09:57

OK, you can now call GetCharCount() for String

Mirek

Great!
Thanks,
Luigi

Subject: Re: String.GetCount with umlaut
Posted by forlano on Wed, 07 Sep 2011 09:11:14 GMT
View Forum Message <> Reply to Message

mirek wrote on Wed, 07 September 2011 09:57

OK, you can now call GetCharCount() for String

Mirek

I wonder if
NFormat(" %-40.40s", s)
should accomodate the string by counting the characters or the bytes of the string. From previous example it seems to work with byte and fails the aligning with accented chars that are multibytes.

Luigi

Subject: Re: String.GetCount with umlaut
Posted by mr_ped on Thu, 08 Sep 2011 07:50:23 GMT
View Forum Message <> Reply to Message

I'm afraid you are opening can of worms right now...
(and it's probably right time to open it, as Mirek is collecting those things for ToDo list)

Subject: Re: String.GetCount with umlaut
Posted by forlano on Thu, 08 Sep 2011 19:17:09 GMT

View Forum Message <> Reply to Message

mr_ped wrote on Thu, 08 September 2011 09:50I'm afraid you are opening can of worms right now...
(and it's probably right time to open it, as Mirek is collecting those things for ToDo list)

Ops... I am sorry, but NFormatChar() would be enough to close the can

Subject: Re: String.GetCount with umlaut
Posted by mirek on Fri, 09 Sep 2011 08:20:00 GMT

View Forum Message <> Reply to Message

forlano wrote on Thu, 08 September 2011 15:17mr_ped wrote on Thu, 08 September 2011 09:50I'm afraid you are opening can of worms right now...
(and it's probably right time to open it, as Mirek is collecting those things for ToDo list)

Ops... I am sorry, but NFormatChar() would be enough to close the can

Nah, I guess it should work in all cases, means with normal Format... (Note that NFormat = Format, since about 4 years back

Mirek

Subject: Re: String.GetCount with umlaut
Posted by mirek on Fri, 09 Sep 2011 10:43:21 GMT

View Forum Message <> Reply to Message

Format now supports utf-8 for padding...

Mirek

Subject: Re: String.GetCount with umlaut
Posted by forlano on Fri, 09 Sep 2011 12:08:35 GMT

View Forum Message <> Reply to Message

mirek wrote on Fri, 09 September 2011 12:43Format now supports utf-8 for padding...

Mirek

Subject: Re: String.GetCount with umlaut
Posted by forlano on Thu, 15 Sep 2011 07:32:00 GMT

View Forum Message <> Reply to Message

mirek wrote on Fri, 09 September 2011 12:43Format now supports utf-8 for padding...
Mirek

The new Format() works perfectly and it is a pleasure to see the text aligned full of accented chars (to make me happy I need little things).
As Mr_Ped said in future would be nice to have some other functions working with chars instead of bytes. One of these function to be put in the top list is Mid(). Sometimes one cut string in the middle of a two bytes character and this produces a mess in qtf viewer. Unfortunately one cannot foresee in advance where the accented char is. In the meanwhile I'll try to convert to WString.

Luigi

Subject: Re: String.GetCount with umlaut
Posted by mirek on Thu, 15 Sep 2011 09:38:58 GMT

View Forum Message <> Reply to Message

forlano wrote on Thu, 15 September 2011 03:32mirek wrote on Fri, 09 September 2011 12:43Format now supports utf-8 for padding...
Mirek

The new Format() works perfectly and it is a pleasure to see the text aligned full of accented chars (to make me happy I need little things).
As Mr_Ped said in future would be nice to have some other functions working with chars instead of bytes. One of these function to be put in the top list is Mid(). Sometimes one cut string in the middle of a two bytes character and this produces a mess in qtf viewer. Unfortunately one cannot foresee in advance where the accented char is. In the meanwhile I'll try to convert to WString.

WString is the preferred way here...

Format was the only thing worth fixing, as it sits in "grey area" between WString and String way...

Mirek