Subject: encoding of command line arguments
Posted by zsolt on Tue, 07 Mar 2006 22:35:20 GMT
View Forum Message <> Reply to Message

CommandLine() function returns a Vector with command line arguments in system enconing.
Other parts of U++ uses UTF-8.
Maybe it could be a good idea to convert encoding in SplitCmdLine__() function.
Currently, I'm using a workaround like:
FromSystemCharset(cmd[i])

Subject: Re: encoding of command line arguments
Posted by mirek on Wed, 08 Mar 2006 08:04:18 GMT
View Forum Message <> Reply to Message

zsolt wrote on Tue, 07 March 2006 17:35CommandLine() function returns a Vector with command line arguments in system enconing. Other parts of U++ uses UTF-8.
Maybe it could be a good idea to convert encoding in SplitCmdLine__() function.
Currently, I'm using a workaround like:
FromSystemCharset(cmd[i])

I guess you are right...

BTW, it is not limited to UTF-8 - that is just the default setting (curiously, not the one we are using in many of our production apps because of historical reasons). You can set it to any encoding you need using SetDefaultCharset.

Mirek

Subject: Re: encoding of command line arguments
Posted by mirek on Sat, 11 Mar 2006 08:08:56 GMT
View Forum Message <> Reply to Message

luzr wrote on Wed, 08 March 2006 03:04zsolt wrote on Tue, 07 March 2006 17:35CommandLine() function returns a Vector with command line arguments in system enconing. Other parts of U++ uses UTF-8.
Maybe it could be a good idea to convert encoding in SplitCmdLine__() function.
Currently, I'm using a workaround like:
FromSystemCharset(cmd[i])

I guess you are right...

BTW, it is not limited to UTF-8 - that is just the default setting (curiously, not the one we are using in many of our production apps because of historical reasons). You can set it to any encoding you need using SetDefaultCharset.

Mirek

After further investigation, I ruled against this, at least for the moment - it is quite unclear when that conversion should happen to keep it safe (a sort of interface shortcoming).

Mirek
