Subject: Making support for code pages unlimited
Posted by Mindtraveller on Sat, 12 Sep 2009 07:56:26 GMT
View Forum Message <> Reply to Message

Recently I`ve came to need of U++ ability to encode bytes into different encodings. Actually, to ANY current encoding available (these bytes could be text in ANY language and ANY encoding).
As fas as I know U++ supports widening it`s internal supported encodings list thus supporting rather limited set of encodings by default.
So if I understand situation correctly adding more encodings into U++ would be handy. I hope to spend some time to make a kind of parser which will take iconv sources' tables and convert them into native U++ format. Does it make sense?

Subject: Re: Making support for code pages unlimited
Posted by mirek on Sat, 12 Sep 2009 14:27:19 GMT
View Forum Message <> Reply to Message

Mindtraveller wrote on Sat, 12 September 2009 03:56Recently I`ve came to need of U++ ability to encode bytes into different encodings. Actually, to ANY current encoding available (these bytes could be text in ANY language and ANY encoding).
As fas as I know U++ supports widening it`s internal supported encodings list thus supporting rather limited set of encodings by default.
So if I understand situation correctly adding more encodings into U++ would be handy. I hope to spend some time to make a kind of parser which will take iconv sources' tables and convert them into native U++ format. Does it make sense?

Yes. The native U++ format is a word table mapping 128 upper characters (< 128 is expected to be ASCII - that means some weird old encoding cannot be supported) to UNICODE.

Mirek

Subject: Re: Making support for code pages unlimited
Posted by Mindtraveller on Sat, 12 Sep 2009 23:38:48 GMT
View Forum Message <> Reply to Message

OK, first step is complete.
1. Filtered almost empty charsets and charsets with more than 2-byte unicode per character (seems like currently unsupported by U++ within common CHARSET_* and CHRTAB_* based internal functions).
2. Codepage names are taken from iconv, which is de-facto standard IMO (previous names could be left for backward compatibility).
3. Parsed and collected data into 2 files (2 pieces each), which are release candidates for insertion into Charset.cpp/.h.

Please look at these files. IMO generally they are good but some charsets could be filtered too.

UPD: Sorry, U++ forum fails to add attachments (sometimes more than one, sometimes more than zero). Will attach files ASAP.

Subject: Re: Making support for code pages unlimited
Posted by Mindtraveller on Sat, 12 Sep 2009 23:42:40 GMT
View Forum Message <> Reply to Message

UPD2: Uploaded files to some file sharing resource:
http://www.2shared.com/file/7757007/3819137b/CharSeth.html
http://www.2shared.com/file/7757008/a8a60eea/CharSetcpp.html

Sorry for using external resources, but forum still denies my attachments (maybe a kind of automatic anti-flood system spoils the game).

Subject: Re: Making support for code pages unlimited
Posted by Mindtraveller on Tue, 15 Sep 2009 20:09:42 GMT
View Forum Message <> Reply to Message

Mirek, could you please look into proposed sources and tell if I need to add anything more to make these codepages added to U++.
(Maybe as some bazaar extension?)
Also I`d like to add standard multibyte code pages for hieroglyphic languages like Japanese, Chinese, etc. Is it possible? I may widen my parser to convert these arrays from iconv sources.

Subject: Re: Making support for code pages unlimited
Posted by andrei_natanael on Wed, 16 Sep 2009 16:14:31 GMT
View Forum Message <> Reply to Message

Files attached.

File Attachments
1) CharSet.h.i, downloaded 431 times
2) CharSet.cpp.i, downloaded 361 times

Subject: Re: Making support for code pages unlimited
Posted by Mindtraveller on Wed, 16 Sep 2009 18:55:41 GMT
View Forum Message <> Reply to Message

Thanks a lot! Strange forum still denies my attaching any files here.

Subject: Re: Making support for code pages unlimited
Posted by mirek on Thu, 17 Sep 2009 15:21:38 GMT
View Forum Message <> Reply to Message

Mindtraveller wrote on Wed, 16 September 2009 14:55Thanks a lot! Strange forum still denies my attaching any files here.

Thanks, this is a wonderful addition!

There were two minor problems, one solved (I have removed some duplicities), one persists - that third parameter of AddCharSetE should be charset that best represents charset on host platform.

In reality, it is perhaps not really useful... but obviously your settings are wrong

Mirek

Subject: Re: Making support for code pages unlimited
Posted by Mindtraveller on Thu, 17 Sep 2009 17:13:17 GMT
View Forum Message <> Reply to Message

luzr wrote on Thu, 17 September 2009 19:21
In reality, it is perhaps not really useful... but obviously your settings are wrong

Mirek
What settings do you mean?

As I understand, you accepted my addition of charsets, and I want to ask if there is a possibility to add multibyte character sets (like Chinese or Japanese) to make supported character sets list complete. I could parse and add them too, but I don't know how to add them to U++.

Subject: Re: Making support for code pages unlimited
Posted by mirek on Thu, 17 Sep 2009 17:24:57 GMT
View Forum Message <> Reply to Message

Mindtraveller wrote on Thu, 17 September 2009 13:13luzr wrote on Thu, 17 September 2009 19:21
In reality, it is perhaps not really useful... but obviously your settings are wrong

Mirek
What settings do you mean?

#ifdef PLATFORM_WIN32
....
 AddCharSetE("iso8859-1",   CHRTAB_ISO8859_1, CHARSET_WIN1252);

Here the last parameter, CHARSET_WIN1252, says that ISO8859_1 equivalent in Win32 is WIN1252 - it mostly contains same characters, although not at same codepoint.

Well, in fact, I think we can happily remove this info... I will check soon if that is possible.

Quote:
As I understand, you accepted my addition of charsets, and I want to ask if there is a possibility to add multibyte character sets (like Chinese or Japanese) to make supported character sets list complete. I could parse and add them too, but I don't know how to add them to U++.

Well, that will be tricky. I think we will have to change charset.cpp internals a bit to support them.

Also, I am not sure that I want a copy of big CJK conversion tables in each application. Maybe this could be in another package (of course, somehow registering into regular charset.h API).

Subject: Re: Making support for code pages unlimited
Posted by Mindtraveller on Thu, 17 Sep 2009 18:18:23 GMT
View Forum Message <> Reply to Message

You are right of course. I thought about that and came to conclusion that the best approach will be making some compile flags (defines, I mean) that will enable compiling additional charsets for those programs which need them.

Subject: Re: Making support for code pages unlimited
Posted by mirek on Sun, 20 Sep 2009 07:20:41 GMT
View Forum Message <> Reply to Message

Mindtraveller wrote on Thu, 17 September 2009 14:18You are right of course. I thought about that and came to conclusion that the best approach will be making some compile flags (defines, I mean) that will enable compiling additional charsets for those programs which need them.

What is wrong with another package?

Subject: Re: Making support for code pages unlimited
Posted by Mindtraveller on Sun, 20 Sep 2009 11:17:39 GMT
View Forum Message <> Reply to Message

If it is possible to make explicit package with these tables defined, then no problem.

Subject: Re: Making support for code pages unlimited
Posted by mirek on Sun, 20 Sep 2009 21:07:37 GMT
View Forum Message <> Reply to Message

Problem:

ARMSCII

and

CP1161

(U++ crashes on ASSERT in debug mode).

I have commented them out for now...

Mirek

Subject: Re: Making support for code pages unlimited
Posted by Mindtraveller on Tue, 22 Sep 2009 07:58:24 GMT
View Forum Message <> Reply to Message

What is the problem with these tables? How could they possibly lead to crash?

Will you make these tables an additional package as planned?

Also I'd like to ask you to change charset.cpp internals for multibyte charsets support. Then I will port to U++ additional ISO codepages for languages like Japanese or Chinese. This will make U++ support for character pages complete.

These will be handy for those who make really widely used apps with U++.

Subject: Re: Making support for code pages unlimited
Posted by mirek on Thu, 24 Sep 2009 08:35:37 GMT
View Forum Message <> Reply to Message

I am sorry I did not have to look into crashing tables more, now I did.

The problem with these crashes was mostly artificial, there were additional check to debug problems in tables:

- check that none of characters in the table is <128 (problem in ARMSCII_8)

- check that there are no duplicates (CP1161).

After removing the check, everything seems to be OK now.

As for multibyte character sets....

There is sort of problem, because some of charset.h expect single character.

So I guess all we can do is some sort of hook into 'whole string' functions that gets extended/reimplemented in "MBCS" package.

Maybe something like:

```
void RegisterMBCS(byte charset,
                  WString (*tounicode)(const char *s, int len),
                  String (*fromunicode)(const wchar *s, int len));
```

or maybe rather

```
void RegisterMBCS(byte charset,
                  WString (*tounicode)(const char *s, int len, int charset),
                  String (*fromunicode)(const wchar *s, int len, int charset));
```

or even

```
void RegisterMBCS(byte charset, void *param,
                  WString (*tounicode)(const char *s, int len, void *param),
                  String (*fromunicode)(const wchar *s, int len, void *param));
```

Mirek

Subject: Re: Making support for code pages unlimited
Posted by Mindtraveller on Thu, 24 Sep 2009 08:47:18 GMT
View Forum Message <> Reply to Message

As for MBCS support, the first option seems to me the most simple. As soon as these functions are implemented (simple axample of usage will be handy), I'll finish quasi-parser and convert remaining iconv tables into U++.
More important question IMO is if it will be possible to use these charsets in common U++ charset coversion functions.

Subject: Re: Making support for code pages unlimited
Posted by mirek on Thu, 24 Sep 2009 09:05:44 GMT
View Forum Message <> Reply to Message

Mindtraveller wrote on Thu, 24 September 2009 04:47As for MBCS support, the first option seems to me the most simple. As soon as these functions are implemented (simple axample of usage will be handy), I'll finish quasi-parser and convert remaining iconv tables into U++.
More important question IMO is if it will be possible to use these charsets in common U++ charset coversion functions.

Read above. Not in all.

E.g.:

int IsLetter(int c, byte chrset);

is obviously impossible with MBCS...

Subject: Re: Making support for code pages unlimited
Posted by Mindtraveller on Thu, 24 Sep 2009 09:46:11 GMT
View Forum Message <> Reply to Message

I mean functions like ToUnicode(), FromUnicode().

Subject: Re: Making support for code pages unlimited
Posted by mirek on Thu, 24 Sep 2009 09:49:07 GMT
View Forum Message <> Reply to Message

Mindtraveller wrote on Thu, 24 September 2009 05:46I mean functions like ToUnicode(), FromUnicode().

Uh, I guess that above proposed API is just for this...

Mirek

Subject: Re: Making support for code pages unlimited
Posted by mirek on Thu, 24 Sep 2009 09:50:05 GMT
View Forum Message <> Reply to Message

Anyway, I guess the hard part here is to implement some MbcsToUnicode and MbcsFromUnicode, I can glue it to charset.h easily myself...

Subject: Re: Making support for code pages unlimited
Posted by Mindtraveller on Thu, 24 Sep 2009 20:05:00 GMT
View Forum Message <> Reply to Message

Thank you. Is there a possibility to add support of newly added charsets and MBCS into QTF?

Subject: Re: Making support for code pages unlimited
Posted by mirek on Fri, 25 Sep 2009 08:28:46 GMT
View Forum Message <> Reply to Message

Mindtraveller wrote on Thu, 24 September 2009 16:05Thank you. Is there a possibility to add support of newly added charsets and MBCS into QTF?

There is always a possibility, but in this particular case I do not see much sense...

After checking QTF code:

Maybe adding something like

int FetchChar(char &* ptr);

- reads multibyte character from source pointer, moves pointer as neede

would be helpful. Or I can change the code a bit more and just use block conversion (from above api).

Subject: Re: Making support for code pages unlimited
Posted by Mindtraveller on Fri, 25 Sep 2009 14:56:04 GMT
View Forum Message <> Reply to Message

OK, Mirek. May be you are right. Then, I consider converting text into UTF-8 and after that into QTF.

Subject: Re: Making support for code pages unlimited
Posted by Mindtraveller on Thu, 01 Oct 2009 21:21:08 GMT
View Forum Message <> Reply to Message

Recently I tried to use newly added charset:

which should had displayed Russian characters inside Windows' console app (which uses native CP866 for characters). But my output was a number of "error" symbols.

I don't know if it is a bug or I did something wrong. How could I convert my UTF8 string into CP866?

Also I'd like to ask Mirek to give a simple example how could I add MBCS charset into U++.

Subject: Re: Making support for code pages unlimited
Posted by mirek on Sat, 03 Oct 2009 19:11:16 GMT
View Forum Message <> Reply to Message

Mindtraveller wrote on Thu, 01 October 2009 17:21Recently I tried to use newly added charset:

which should had displayed Russian characters inside Windows' console app (which uses native CP866 for characters).

Maybe it does not.

IMO the simple way is to create a testcase that stores it into the file, then view the file. If it does not work, you will have a testcase for me:)

Quote:
Also I'd like to ask Mirek to give a simple example how could I add MBCS charset into U++.

I guess we agreed to "make routines to convert To and From Unicode wchar *, len and I will add it to U++".... ?

Mirek

Subject: Re: Making support for code pages unlimited
Posted by Mindtraveller on Tue, 09 Feb 2010 22:45:38 GMT
View Forum Message <> Reply to Message

Recently I had some time to analyze encodings. And I've met a number of issues to be discussed.

First, I discovered that CHARSET_**** tables are not in unicode, but in UTF-8. If it is so, I should rebuild proposed tables according to this.

Second issue is simple etude, which actually failed. Just look at this simple code:CONSOLE_APP_MAIN
{

}

I executed this example as console app under Windows XP. My "native" console code page under XP is CP866. But on program run, instead of cyrillic letters, I've seen garbage symbols. My question: what is wrong in this example and why do you think it fails to convert symbols properly?

P.S. I tried to update CHARSET_CP866 array to contain UTF-8 encoded symbols, but this little example still fails to convert symbols propely.

Subject: Re: Making support for code pages unlimited
Posted by mirek on Sat, 13 Feb 2010 12:58:40 GMT
View Forum Message <> Reply to Message

Mindtraveller wrote on Tue, 09 February 2010 17:45Recently I had some time to analyze encodings. And I've met a number of issues to be discussed.

First, I discovered that CHARSET_**** tables are not in unicode, but in UTF-8. If it is so, I should rebuild proposed tables according to this.

What makes you think that? Those tables are just UTF-16 codes for characters 128-255.

Quote:
Second issue is simple etude, which actually failed. Just look at this simple code:CONSOLE_APP_MAIN
{

}
I executed this example as console app under Windows XP. My "native" console code page under XP is CP866. But on program run, instead of cyrillic letters, I've seen garbage symbols. My question: what is wrong in this example and why do you think it fails to convert symbols properly?

Well, it looks like things are more complicated. I believe our part is OK, but the console simply expcts the output to be in different charset. E.g.:

http://illegalargumentexception.blogspot.com/2009/04/i18n-unicode-at-windows-command-prompt.html

However, more thinking about it, I believe we should perhaps use Unicode variant of WriteFile and convert current app encoding (which is utf8 by default) to unicode.

In that case, however, you example would work without ToCharset...

Mirek

Subject: Re: Making support for code pages unlimited
Posted by Mindtraveller on Sun, 14 Feb 2010 00:15:11 GMT
View Forum Message <> Reply to Message

luzr wrote on Sat, 13 February 2010 15:58I believe our part is OK, but the console simply expcts the output to be in different charset.I believe it's not. This code

```
 Cout() << "\n";

 for (int i=0; i<s.GetLength(); ++i)
  Cout() << Format("%02X ", s[i]);
```

gives output:Quote:1F 1F 1F 1F 20 1F 1F 1F 1F 1F 1F 21
Looks like our problem.