Subject: about the using of unicode in win32 platform
Posted by LeiMing on Thu, 07 Feb 2013 04:40:51 GMT
View Forum Message <> Reply to Message

today I'm thinking about how to deal with commandline parameter,
and finding that ultimate++ currently using WinMain with LPSTR lpCmdLine parameter.

I think it's better to using unicode via wWinMain with LPWSTR lpCmdLine parameter. I change the macro a little, becoming

C:\upp\uppsrc\CtrlCore\Win32GuiA.h
```
#define GUI_APP_MAIN \
void GuiMainFn_();\
\
int APIENTRY wWinMain(HINSTANCE hInstance, HINSTANCE, LPWSTR lpCmdLine, int nCmdShow) \
{ \
 UPP::Ctrl::InitWin32(hInstance); \
 UPP::coreCmdLine__() = UPP::SplitCmdLine__(UPP::ToUtf8(lpCmdLine)); \
 UPP::AppInitEnvironment__(); \
 UPP::AppExecute__(GuiMainFn_); \
 UPP::Ctrl::CloseTopCtrls(); \
 UPP::Ctrl::ExitWin32(); \
 UPP::AppExit__(); \
 return UPP::GetExitCode(); \
} \
\
void GuiMainFn_()
```

It works fine for me.
Do you consider about this? I think it's a improvement in internalization.

by the way, is there any official way to get access to argument of execute commandline? I don't know if coreCmdLine__ is the interface for library user.

regards,
LeiMing

Subject: Re: about the using of unicode in win32 platform
Posted by dolik.rce on Thu, 07 Feb 2013 06:09:51 GMT
View Forum Message <> Reply to Message

LeiMing wrote on Thu, 07 February 2013 05:40by the way, is there any official way to get access to argument of execute commandline? I don't know if coreCmdLine__ is the interface for library user.
There is CommandLine() function, which returns Vector containing the command line arguments, stripped of the zeroth argument (the executable name, which has its own method GetExeTitle()).

Also this method is aware of charset, so probably the utf8 should be handled in there rather than in wWinMain.

Best regards,
Honza

Subject: Re: about the using of unicode in win32 platform
Posted by LeiMing on Tue, 12 Feb 2013 15:40:05 GMT
View Forum Message <> Reply to Message

since windows has the function CreateProcessW, it's necessary to handle parameters in unicode: WinMain has commandline encoded in local charset, which may not be able to present some chars: they can be out of current charset. While wWinMain can properly handle the situation: the parameters are stored in wchar_t[].

Subject: Re: about the using of unicode in win32 platform
Posted by mirek on Tue, 12 Feb 2013 17:05:05 GMT
View Forum Message <> Reply to Message

LeiMing wrote on Tue, 12 February 2013 10:40since windows has the function CreateProcessW, it's necessary to handle parameters in unicode: WinMain has commandline encoded in local charset, which may not be able to present some chars: they can be out of current charset. While wWinMain can properly handle the situation: the parameters are stored in wchar_t[].

Well, I would say it is about time to finally drop last bits of Win98 support

Mirek
