

Re: add support for other languages

Source: <http://www.tech-archive.net/Archive/VC/microsoft.public.vc.mfc/2005-06/msg02070.html>

- *From:* Joseph M. Newcomer <newcomer@xxxxxxxxxxxxx>
 - *Date:* Wed, 29 Jun 2005 17:28:59 -0400
-

If you have a STRINGTABLE suitably set up for internationalization (either different language DLLs or a single DLL with multiple language support, and use LoadString to load the string.

MessageBox does not use single-byte LPCSTR; it definitely is specified as taking LPCTSTR. Read the documentation. So that isn't even an issue to deal with. Just compile in Unicode mode and the problem is solved!

The more serious problem is the defects in Visual Studio. For example, instead of fixing REAL problems in Visual Studio, some nitwit went off and destroyed its usability. But VS 2003 cannot edit Unicode STRINGTABLE resources (which would have been a worthwhile addition). It shows how little grasp this guy had on what is and is not important to programmers.

You can create a hand-edited resource that contains Unicode strings, but you have to do it as L"\xXXXX\xXXXX\xXXXX" where XXXX is the hex code for the character. As long as you don't edit this with VS, you should be safe. Overall, Microsoft has been incredibly behind the curve in producing useful tools for us to construct actual internationalized apps. (Right now, I have a text editor that writes its output as UTF8 so I can edit it with ordinary editors. So that's another option. Store it as UTF8 and after doing a LoadString, to a MultiByteToWideChar call to convert it to Unicode).

I've heard a rumor that Unicode source files and Unicode resource files are supported in VS2005 ("Whidby"), but I haven't installed that yet (I don't yet have a spare machine to install it on, although that is changing; my new testbed machine arrived a couple hours ago).

There's a microsoft.public.win32.programmer.international that is probably going to be more help on this.

joe

On Wed, 29 Jun 2005 14:35:04 +0100, "Chris Ward" <professorcad@xxxxxxxxxxxxx> wrote:

>I have a tiny program that installs add-on modules for use by our main
>application. The installer basically goes about it's business silently, but
>I have a Windows MessageBox pop-up to report errors (only if encountered). I
>would like to have those error messages in several different languages.

Re: add support for other languages

>Currently, the installer reads the messages from a simple ASCII text file,
>and so the language scope is very limited, however the advantage is that the
>translator can simply swap my file for his and the installer uses his
>messages without me even having to see anything.
>
>MessageBox uses single-byte LPCSTR, so that is the first interesting problem
>to overcome. However, what is more interesting is the file that contains the
>strings in the first place. I can send an English-language plain text file
>to the translators – but what kind of file can they store their language
>versions of the strings in, and how does one parse this file – for example,
>it is "easy" to use MS Word to create sentences in Chinese Simplified, but
>that file would of course contain more data than just the text. Has anybody
>tackled these issues?
>
>regards
>
>IdeasVacuum
>
>
>"Joseph M. Newcomer" <newcomer@xxxxxxxxxxxx> wrote in message
>news:gsoub15hd819mjrul1tja3bds1bate38dks@xxxxxxxxxx
>> There are a lot of issues dealing with internationalization. They include
>> simple text
>> (menus, captions, etc.), presentation to the user of text (e.g., a
>> dynamically-loaded
>> ComboBox needs to use language-specific strings, and you cannot make
>> assumptions about the
>> ordering if it is sorted by string name. The number of people who use the
>> design time
>> "combo box data" option, one of the most completely useless options ever
>> added to a
>> development environment, is frightening. It rarely, if ever, is useful.
>> Then you've got
>> details about how to represent dates, numbers, times, etc. Some of these
>> decisions are
>> context-sensitive, e.g., some countries use 12-hour time for "civilian"
>> uses but
>> applications designed to run in contexts such as transport scheduling are
>> required to use
>> 24-hour times. Right-to-left and left-to-right displays make life more
>> interesting (Hebrew
>> and Arabic are currently giving me grief, particularly because I cannot
>> read either
>> alphabet).
>>
>> You have to deal with the fact that the controls may need to be resized to
>> handle the
>> alternate language strings (apparently Finnish is the worst language,
>> because the strings
>> can have up to 3x the number of characters of English strings).
>>

Re: add support for other languages

Re: add support for other languages

>> You have to deal with locale-specific sort order (German has two, for
>> example, default and
>> telephone book, and no, I don't know the difference).
>>
>> If you don't mind playing with pre-alpha code, send me some email and I'll
>> send you a
>> snapshot of the program I'm working on to help people who deal with these
>> issues, my
>> "Locale Explorer". I'm still adding features to it, so not all features
>> are fully
>> implemented, and may not give any output.
>>
>> First step is to remove every single native-language string (except
>> possibly your own
>> personal debug output messages that will not appear in the delivered
>> product) from the
>> source code. Second step, which is easy to do at the same time, is to make
>> everything
>> Unicode-aware. For example, 99.999% of the uses of 'char' are erroneous
>> for a program that
>> is Unicode-aware. So you must be incredibly suspicious of every appearance
>> of 'char',
>> LPSTR, LPCSTR, and other 8-bit assumptions. You cannot assume characters
>> are single bytes.
>> Every literal must be surrounded by _T() and the exceptions to this are so
>> vanishingly
>> small that you can, as a first approximation, assume that any such usage
>> is erroneous.
>>
>> Now, if you have external data, such as a file, a communication line,
>> etc., then you have
>> to worry about converting locale-specific characters to and from the
>> communication
>> representation. Some communication links talk to programs that can only
>> support 7-bit
>> characters, so worrying about how you deal with this when there are
>> accented characters
>> can pose interesting problems (sure, UTF7 solves the representation
>> problem, but only if
>> the OTHER side understands UTF7 also!)
>>
>> I don't even want to get into UniScribe. I'm working out ways of avoiding
>> its use entirely
>> (this is not easy, but relevant for several classes of languages,
>> including mixed RTL and
>> LTR text).
>>
>> So it depends on what you are trying to deal with as to how hard your
>> problem will be.
>> Right now, I can handle a variety of scripts, although I am not certain I
>> have a clue as

Re: add support for other languages

Re: add support for other languages

>> to whether or not the output is actually completely valid (Chinese numbers
>> are a problem
>> I've shelved for a while, because there is a symbol very similar in
>> appearance to a + that
>> represents "10", and I just haven't had time to figure out what to do here
>> (once I put
>> this out with complete source, I will be happy to accept corrections, but
>> it is Not Yet
>> Ready For Primte Time until I go as far as I can on my own).
>>
>> Overall, internationalization raises some fascinating issues, and there is
>> a series of
>> MSDN articles on how to deal with it.
>>
>> Start with the articles at
>>
>> <http://www.microsoft.com/globaldev/default.msp>
>>
>> and follow the links around. I'm still absorbing all the material here.
>> joe
>> On Sat, 25 Jun 2005 21:26:50 +0200, "Sam" <sam@xxxxxxxxxx> wrote:
>>
>>>Dear Dave,
>>>
>>>I'm sorry but I was misleading a little. My problem is with the different
>>>languages. I don't know where is the language support comes into
>>>expression – in the compiler(I have only VC in English), in O/S or
>>>elsewhere. For instance I live in Israel, I create applications in my PC
>>>which its default language is Hebrew and I never get Hebrew whatsoever.
>>>
>>>I hope I clarified a little.
>>>
>>>Regards
>>>
>>>Sam
>>>
>>>"David Lowndes" <davidl@xxxxxxxxxxxxxxxx> wrote in message
>>>news:9uoqb15dbbke1i9rot083aftg6ddke67nh@xxxxxxxxxx
>>>> >I think I'll cope with that, but how to create these dll's.
>>>> >Can you refer to a link or documentation also?
>>>>
>>>> Sam,
>>>>
>>>> Try this:
>>>>
>>>>
>>>>http://msdn.microsoft.com/library/en-us/vccore/html/_core_Create_a_Resource_2d.Only_DLL.asp
>>>>
>>>> Dave
>>>> --

Re: add support for other languages

Re: add support for other languages

>>>> MVP VC++ FAQ: <http://www.mvps.org/vcfaq>
>>>
>>
>> Joseph M. Newcomer [MVP]
>> email: newcomer@xxxxxxxxxxxx
>> Web: <http://www.flounder.com>
>> MVP Tips: http://www.flounder.com/mvp_tips.htm
>

Joseph M. Newcomer [MVP]
email: newcomer@xxxxxxxxxxxx
Web: <http://www.flounder.com>
MVP Tips: http://www.flounder.com/mvp_tips.htm
.

- **Follow-Ups:**
 - ◆ **[Re: add support for other languages](#)**
◇ From: David Lowndes

- **References:**
 - ◆ **[add support for other languages](#)**
◇ From: Sam
 - ◆ **[Re: add support for other languages](#)**
◇ From: David Lowndes
 - ◆ **[Re: add support for other languages](#)**
◇ From: Sam
 - ◆ **[Re: add support for other languages](#)**
◇ From: David Lowndes
 - ◆ **[Re: add support for other languages](#)**
◇ From: Sam
 - ◆ **[Re: add support for other languages](#)**
◇ From: Joseph M . Newcomer
 - ◆ **[Re: add support for other languages](#)**
◇ From: Chris Ward

- Prev by Date: **[Re: MFC design question](#)**
- Next by Date: **[Re: afxwin1.inl – Debug Assertion Failed](#)**
- Previous by thread: **[Re: add support for other languages](#)**
- Next by thread: **[Re: add support for other languages](#)**
- Index(es):
 - ◆ **[Date](#)**
 - ◆ **[Thread](#)**