If the system says that an embedded string could not be converted from Unicode to ANSI, maybe it's trying to tell you something

devblogs.microsoft.com/oldnewthing/20070703-00

July 3, 2007

Raymond Chen

It's probably trying to tell you that an embedded string could not be converted from Unicode to ANSI.

> One of our programs is throwing the exception "Type could not be marshaled because an embedded string could not be converted from Unicode to ANSI." It happens only if we use the Chinese version of the program. Why are we getting this exception?

I may be going out on a limb here, but I bet it's because an embedded string could not be converted from Unicode to ANSI. Unicode is big. Bigger than any ANSI code page. No matter what ANSI code page you pick, there will be Unicode characters that cannot be expressed in it. (And no, you can't set your ANSI code page to UTF-8. Michael Kaplan discussed it last October, and before that, last July, and before that, a week and a half previous (still July), and before that, two years ago February. I think Michael might need to change the subtitle of his blog to "Explaining why the ANSI code page can't be UTF-8 since 2005".) In particular, if you grab some Chinese characters and try to map them to the ANSI code page on an English system, they definitely won't map successfully. The real question, though, is why you're using ANSI in the first place. Get with the program, it's 2007 already. As Michael Kaplan noted, it's all Unicode all the time from now on. In that same article from Michael, he also does the necessary psychic debugging for the root cause: The `DllImport` declaration did not specify a `CharSet`, so you get the default, which is `CharSet.Ansi`!

Change your character set to `CharSet.Unicode` and you'll be all set.

Raymond Chen

Follow