How do I convert a UTF-8 string to UTF-16 while rejecting illegal sequences?

devblogs.microsoft.com/oldnewthing/20110615-00

June 15, 2011

Raymond Chen

By default, when you ask `MultiByteToWideChar` to convert a UTF-8 string to UTF-16 that contains illegal sequences (such as overlong sequences), it will try to muddle through as best as it can. If you want it to treat illegal sequences as an error, pass the `MB_ERR_INVALID_CHARS` flag.

The MSDN documentation on this subject is, to be honest, kind of hard to follow and even includes a double-negative: "The function does not drop illegal code points if the application does not set this flag." Not only is this confusing, it doesn't even say what happens to illegal code points when you omit this flag; all it says is what it *doesn't* do, namely that it doesn't drop them. Does it set them on fire? (Presumably, if you omit the flag, then it retains illegal code points, but how do you retain an illegal UTF-8 code point in UTF-16 output? It's like saying about a function like `atoi` "If the value cannot be represented as an integer, it is left unchanged." Huh? The function still has to return an integer. How do you return an unchanged string as an integer?)

Raymond Chen

Follow