Feel free to stop using IMultiLanguage2::DetectInputCodepage

devblogs.microsoft.com/oldnewthing/20221003-00

October 3, 2022

Raymond Chen

A customer reported that they were observing some weird behavior when passing the text of an email message to `IMultiLanguage2::DetectInputCodepage`. They were passing the `MLDETECTCP_HTML` flag, claiming that I recommended passing that flag, but found that using that flag caused `DetectInputCodepage` to detect the wrong code page, whereas if they passed `MLDETECTCP_NONE`, the detection worked a little better. Not perfectly, but a little better.

Okay, first of all, they appear to have read only the first half of the sentence where I explained when to pass the `MLDETECTCP_HTML` flag:

> One thing that may not be obvious is that the program passes the `MLDETECTCP_HTML` flag if the file extension is `.htm` or `.html`.

Somehow the customer though I was recommending that you always pass the `MLDETECTCP_HTML` flag. No, you pass that flag only if you are passing in HTML. That way, the detector will not consider text inside angle brackets as evidence that the contents are written in English. That text is part of the HTML tags and is not part of the message text.

"The customer found that the `MLDETECTCP_NONE` flag works for them. We want to know if there are any risks of using that flag instead of the `MLDETECTCP_HTML` flag."

The risk is that you might be providing correct information to the `DetectInputCodepage` function and therefore allow it to perform as intended. I guess that's a good risk.

But really, feel free to stop using `IMultiLanguage2::DetectInputCodepage`!

Code page detection is never going to be reliable. The function has always been a guess. An educated guess based on available information but it's still a guess. Guesses can be wrong.

The customer wrote back, "For the attached message, `DetectInputCodepage` returns code page 1252 when passed `MLDETECTCP_NONE`, but when we use code page 1252 to convert the message to Unicode, the results are incorrect. If we explicitly convert with code page 28592, then the results are correct. Is there a workaround for this?"

The workaround is to pass 28592 when the code page is 28592. You can get the code page from other metadata attached to the message, such as a `charset=` attribute in the `Content-Type`. If that metadata is missing, then you are going to be forced to guess, and the nature of guessing is the possibility that you might guess *wrong*.

Raymond Chen

Follow