

Nasty gotcha: SetThreadUILanguage cannot be used to restore the thread UI language

devblogs.microsoft.com/oldnewthing/20170908-00

September 8, 2017



Raymond Chen

Suppose you want to change the thread UI language temporarily. You might be tempted to do this:

```
// Code in italics is wrong
void Something()
{
    // Save the original language.
    LANGID originalLanguage = GetThreadUILanguage();

    // Set a new language temporarily.
    SetThreadUILanguage(newLanguage);

    // ... do stuff that uses the new language ...

    // Restore the original language.
    SetThreadUILanguage(originalLanguage);
}
```

This seems to work, but in fact it doesn't.

The `GetThreadUILanguage` function returns the first user interface language for the current thread. If a preferred language has been set for the thread, it will use that. Otherwise, it will follow a documented fallback algorithm.¹

On the other hand, `SetThreadUILanguage` sets the UI language for the current thread. It never sets the thread language back to “not set”.

In the above code fragment, the result is that the thread UI language is locked to whatever the effective thread UI language was at the time the function was called, even if the fallback languages change.

For example, suppose the user's preferred language is English, the process's preferred language is German, and the thread has no preferred language. The call to `GetThreadUILanguage` will return German, and when the function tries to restore the original language,

it sets the thread's preferred language to German. This is not the same as clearing the thread's preferred language, however. If the process changes its preferred language to Swedish, and the thread has no preferred language, then the effective language should change to Swedish. But the code fragment above explicitly sets the thread language to German, so the effective language will be German.

The way to restore the thread preferred UI language state is to capture the thread preferred UI languages with the `GetThreadPreferredUILanguages` function and restore them with the `SetThreadPreferredUILanguages` function. For more information, [see my earlier discussion](#).

¹ The documentation for [GetThreadUILanguage](#) says

Calling this function is identical to calling `GetThreadPreferredUILanguages` with *dwFlags* set to `MUI_MERGE_SYSTEM_FALLBACK | MUI_MERGE_USER_FALLBACK | MUI_LANGUAGE_ID` and using the first language in the retrieved list.

You can then follow the bouncing ball to [the documentation for GetThreadPreferredUILanguages](#) and read the description of what happens when you ask for both system and user fallbacks.

[Raymond Chen](#)

Follow

