The sad history of the MFC TRY/CATCH macros

devblogs.microsoft.com/oldnewthing/20190911-00

September 11, 2019

Raymond Chen

Why does MFC define `TRY/CATCH` macros? Why can't it use standard C++ exception handling?

This is another case of asking why they didn't use the Space Shuttle to rescue Apollo 13.

MFC was first released in 1992. At that time, C++ was a cool new programming language that everybody was interested in. Back in the early days, there were no templates, there was no standard library, no placement new, no namespaces, and no exceptions.[1]

That meant that the MFC team had to invent their own exceptions, which they did with macros like `TRY` and `CATCH`. These macros were deprecated in MFC 3.0 (released in 1994), but for compatibility the macros continue to exist so that pre-MFC-3.0 code remains source-level compatible. The macros now just do standard C++ `try` and `catch` operations, although there's a `_AFX_OLD_EXCEPTIONS` symbol you can define to make them revert to the old `setjmp`-based fake exception handling, for binary compatibility.

MSDN has a guide to converting from MFC exceptions to C++ exceptions. So convert your code already. You've had 25 years.

[1] This implies that a failed `new` didn't throw a `std::bad_alloc` exception, seeing as neither namespaces nor exceptions existed yet! Instead, a failed `new` simply returned `NULL`.[2]

[2] The `nullptr` keyword wouldn't exist for nearly 20 years.

Raymond Chen

Follow