

Why doesn't Clipboard History capture rapid changes to clipboard contents?

 devblogs.microsoft.com/oldnewthing/20250508-00/?p=111162

May 8, 2025



A customer was trying to write a tool to preload items into the clipboard history. In a way, this is sort of the opposite of history. Instead of looking backward to things that were once on the clipboard in the past, they want to look forward into the future and preload things that they anticipate you are soon going to wish were on the clipboard. Here's a stripped-down version.

```

// All error checking elided for expository purposes
#include <windows.h>

void SetClipboardText(HWND hwnd, PCWSTR text)
{
    OpenClipboard(hwnd);
    EmptyClipboard();
    auto size = sizeof(wchar_t) * (1 + wcslen(text));
    auto clipData = GlobalAlloc(GMEM_MOVEABLE, size);
    auto buffer = (LPWSTR)GlobalLock(clipData);
    strcpy_s(buffer, size, text);
    GlobalUnlock(clipData);
    SetClipboardData(CF_UNICODETEXT, clipData);
    CloseClipboard();
}

// Put these strings in the clipboard history for quick access.
static constexpr PCWSTR messages[] = {
    L"314159", // the bug number we want to edit
    L"e83c5163316f89bfbde7d9ab23ca2e25604af290", // the commit to link the bug to
    L"Widget polarity was set incorrectly.", // the comment to add
};

int wmain([[maybe_unused]] int argc,
          [[maybe_unused]] wchar_t* argv[])
{
    auto tempWindow = CreateWindowExW(0, L"static", nullptr, WS_POPUPWINDOW,
                                       0, 0, 0, 0, nullptr, nullptr, nullptr, nullptr);

    for (auto message : messages)
    {
        SetClipboardText(tempWindow, message);
    }
    DestroyWindow(tempWindow);
    return 0;
}

```

This program sets three strings onto the clipboard one after the other. But when you run it, only the last string makes it into the clipboard history. What happened to the other two?

The clipboard history service operates asynchronously. It registers for clipboard changes via `AddClipboardFormatListener`, and when it receives a change notification, it updates the clipboard history. The listener is notified asynchronously, however, so by the time the listener receives the `WM_CLIPBOARDUPDATE` message, the clipboard may have changed a second time.

[This is different from clipboard viewers](#), which are notified synchronously when the clipboard changes. The downside is that you might miss out on clipboard changes. The much better upside is that you don't slow down or hang the clipboard.

In practice, missing every little clipboard change is sort of a feature of the clipboard history service. I can imagine that there are programs that just spam the clipboard with a rapid sequence of clipboard changes. All of the intermediate ones are useless because

they are never on the clipboard long enough for the user to paste them. Only the last one really counts from an end-user point of view, so it's reasonable that the clipboard history service matches what the user sees on the clipboard.

Next time, we'll see what we can do to repair this program.