

Enumerating Windows clipboard history in PowerShell

 devblogs.microsoft.com/oldnewthing/20230303-00

March 3, 2023



Raymond Chen

Last time, we enumerated the contents of the Windows clipboard history from C++/WinRT and C#. Today we'll do it from PowerShell.

Because, hey, why not.

```
$null = [Windows.ApplicationModel.DataTransfer.Clipboard,
Windows.ApplicationModel.DataTransfer, ContentType=WindowsRuntime]
$op = [Windows.ApplicationModel.DataTransfer.Clipboard]::GetHistoryItemsAsync()

$result = Await ($op) `
    ([Windows.ApplicationModel.DataTransfer.ClipboardHistoryItemsResult])

$textops = $result.Items.Content.GetTextAsync()
for ($i = 0; $i -lt $textops.Count; $i++){ Await($textops[$i]) ([String]) }
```

It's basically the same thing we've been doing, just written in PowerShell. Note that I'm not a PowerShell expert, so some of the things I'm doing may be suboptimal.

First, we load the Clipboard class into memory, specifying that it is a Windows Runtime class.

Next, we call `GetHistoryItemsAsync` and `Await` it to get the history items result.

We take the `Items` from the result, get their `Content`, and ask each `Content` for its text.

The text query is another asynchronous operation, so we iterate through the operations and `Await` each one, sending the results back into the pipeline.

I didn't add the code to check ahead of time whether the content contained text. I just let the exception flow out of the `GetTextAsync` call. Fixing this is left as an exercise.