**FLAUI**

**Features:**

* It’s an Open-source (still now supported)
* It is a .NET library that helps with automated UI testing of Windows applications.
* It is the wrapper around the Microsoft Native UI Automation libraries.
* Supported Language >> C#
* We can take screenshots using Capture class.
* We can take Video of the Execution using VideoRecorder class.
* It also supports Mouse and Keyboard Action
* Assertion is available in Flaui
* Some ideas are copied from the UIAComWrapper project or TestStack.White but rewritten from scratch to have a clean codebase.

**Taking Screenshot using Capture:**

* *Capture.Screen();* >> Takes the entire screen
* *Capture.Element(window)*
* *Capture.Rectangle*
* *Capture.ElementRectangle*

**Taking Video using VideoRecorder:**

sample code:

*var recorder = new VideoRecorder (new VideoRecorderSettings {*

*VideoQuality = 26, ffmpegPath = @"C:\Temp\ffmpeg.exe", TargetVideoPath = @"C:\temp\out.mp4"};*

*});*

**Why Flaui:**

There are quite some automation solutions out there. Commercial ones like TestComplete, Ranorex, CodedUI just to name a few. And also, free ones which are mainly TestStack.White.

All of them are based on what Microsoft provides. These are the UI Automation libraries. There are three versions of it:

MSAA: MSAA is very obsolete and we'll skip this here (some like CodedUI still use it)

UIA2: Managed Library for native UI Automation API

UIA2 is managed only, which would be good for C# but it does not support newer features (like touch) and it also does not work well with WPF or even worse with Windows Store Apps.

UIA3: Com Library for native UI Automation API

UIA3 is the newest of them all and works great for WPF / Windows Store Apps but unfortunately, it can have some bugs with WinForms applications (see FAQ) which are not existent in UIA2.

So, the commercial solutions are mostly based on multiple of those and/or implement a lot of workaround code to fix those issues. TestStack.White has two versions, one for UIA2 and one for UIA3 but because of the old codebase, it's fairly hard to bring UIA3 to work. For this, it also uses an additional library, the UIAComWrapper which uses the same naming as the managed UIA2 and wraps the UIA3 com interop with them (one more source for errors). FlaUI now tries to provide an interface for UIA2 and UIA3 where the developer can choose, which version he wants to use. It should also provide a very clean and modern codebase so that collaboration and further development is as easy as possible.

**Reference:**

* *https://github.com/FlaUI/FlaUI/tree/master/src/FlaUI.Core.UITests*