

Unattended Processing - Application Automation

Asked 15 years, 10 months ago Modified 7 years, 1 month ago

Viewed 933 times



2



I'm looking for information [I hesitate to infer "Best Practices"] for Automating Applications. I'm specifically referring to replacing that which is predictably repeatable through traditional manual means [humans manipulating the GUI] with something that is scheduled by the User and performed "Automatically".

We use AutoIT internally for performing Automated Testing and have considered the same approach for providing Unattended Processing of our applications, but we're reluctant due to the possibility of the user "accidentally" interacting with the Application *in parallel* with the execution of a scheduled "automation" and therefore "breaking" the automation.

Shy of building in our own scheduler with known events and fixed arguments for controlling a predefined set of actions, what approaches should I evaluate/consider and which tools would be required?

Additional Information: Some would refer to this capability as "Batch Processing" within the application context.

[automation](#)[autoit](#)[unattended-processing](#)[Share](#)[Improve this question](#)[Follow](#)

edited Oct 24, 2017 at 23:35

[user4157124](#)

2,904 ● 14 ● 30 ● 44

asked Feb 9, 2009 at 19:07

[SAMills](#)

456 ● 4 ● 13

3 Answers

Sorted by:

Highest score (default)



2



In general it is a hazardous practice to automate UIs. It can be a useful hack for a short term problem: I find myself using AutoHotKey to run some tedious tasks in some situations... but only if the task is not worthy of writing code to implement the change (i.e., a one time, 15 minute task).

Otherwise, you will likely suffer from inconsistent runs due to laggy response of some screens, inconsistent UIs, etc. Most applications have an API available, and not using it is going to be far more painful than acquiring and using it in 99% of cases.

In the unfortunate but possible situation that there is no UI and you are reduced to screen scraping/manipulating, a tool that performs automated testing is probably as good as you will get. It allows you to *verify* the state of the

app (to some degree) and thus can build some safety nets in. Additionally, I would dedicate a workstation to this task... with the keyboard and mouse locked away from curious users. (A remote desktop or VNC style connection works well for this: you can kick off the process and disconnect, making it resistant to tampering.)

However, I would consider that approach only as a desperate last resort. Manipulating an API is far, far, far, far (did I get enough "fars" in there?) more sustainable.

Share Improve this answer

answered Feb 9, 2009 at 19:33

Follow



Godeke

16.3k ● 4 ● 64 ● 87



0



If I understand correctly, you want to do automated *processing* using some tool that will execute a predefined list of actions in a given software system. This being different from automated *testing*.



I would urge you to avoid using tools meant for testing to perform processing. Many major software systems have public APIs you can use to perform actions without direct user interaction. This is a much more robust and reliable way to schedule automated processes. Contact the vendor of the software you are working with, sometimes the API's are available upon request.

Share Improve this answer

answered Feb 9, 2009 at 19:30

Follow



Dave Swersky

34.8k ● 9 ● 82 ● 120



0



Godeke and Dave are absolutely correct that, if available, the API is the best route. However, practically this is sometimes not possible, and you have to go the GUI automation route. In addition to the previously mentioned dedicated workstation(s) to run the automation, I recommend coding in some audit trails, so that it is easier to debug or backtrack if problems arise. Your batch processing automation should keep a detailed log of what records were processed, when they were processed and how they were processed. You should set it up so that the records themselves (in the native application) will reflect that it was updated/processed via automation. For example, if each record has an updateable notes/comments field, the automation should add text to this field like "Processed by automation user, 2009-02-25 10:05:11 AM, Account field changed from 'ABC123' to 'DEF456'" That way, the automated mods will be readily apparent to a user manually pulling up the record in the GUI.

Share Improve this answer

answered Feb 26, 2009 at 14:42

Follow



Tom E

2,482 ● 2 ● 18 ● 22