

Running Automated GUI tests with...



Jason A 2 posts since

Jul 28, 2009

Hello,

I am having a problem with running my GUI tests through TeamCity.

The GUI tests are written with the Autolt scripting tool. I had to use this tool, because our applications are not .NET applications, so I use Autolt's commands like WinActivate, Send, ImageSearch, MouseClick(x,y) etc...

If I logon to the testing server manually and run the tests, everything works fine.

However, if I add the tests to the MSBUILD and then run through TeamCity, all the tests fail because Windows won't allow the process to WinActivate, Send, etc...

Is there a way to get around this?

I have tried being logged in to the virtual machine while the tests are run. But it does not help because the tests are being run in a different session??

I have tried to set the service to run as the local system account, but I think I ran into permission issues when the service tried to delete some files?

Would using VMWare or VNC fix my problem? If so, how would I set this up in TeamCity?

I would be very grateful for any help.

Running Automated GUI tests with...

Thanks,

Jason.

Tags: teamcity, gui, autoit



[Ray Navarette](#) 79 posts since

Dec 7, 2007 1. **Re: Running Automated GUI tests with TeamCity** Jul 29, 2009 4:52 PM

TeamCity doesn't handle UI. My guess is you're running a windows build agent as a service. Windows services have no access to the console though, so this will get you nowhere. To get around this, on the agents where we want to run GUI tests, we run TeamCity in console mode. Simply run agent.bat (in the TC bin folder) with the parameter start (or stop to stop the agent). Of course, you'll have to leave a desktop session open for this to work properly...

Also, to the JetBrains team, this seems to be a growing topic of interest. Think maybe someplace in the docs something like this could be mentioned?

good luck.

ray



[Mark Walker](#) 12 posts since

Jan 11, 2009 2. **Re: Running Automated GUI tests with TeamCity** Jul 30, 2009 6:57 AM

I don't know much about this, but isn't there a "Allow access to the Desktop" or some such property when setting an account to run the service? Would setting that help?



[Carel Lotz](#) 78 posts since

Jul 28, 2008 3. **Re: Running Automated GUI tests with TeamCity** Jul 30, 2009 11:54 AM

Nope, this won't help. See [my post](#) on the topic. We got it working using this process, but it is still very flaky. I'm not sure whether the problem is TeamCity/WebAii related. If we run

Running Automated GUI tests with...

the suite of tests using only WebAii on the same box, they run through. But as soon as we automate the process using TeamCity they fail intermittently. Not ideal 😞



[Ray Navarette](#) 79 posts since

Dec 7, 2007 4. **Re: Running Automated GUI tests with TeamCity** Jul 30, 2009 4:35 PM

Carel,

Do you mean that selecting the allow interaction option for the service doesn't work, or running through console mode doesn't work? They are two very different options. Windows services run in a context with absolutely no console availability on their own. If the allow interaction option is selected, the service may then (under certain circumstances) be able to display a GUI using an already running console (a logged in user's session). I don't believe that TC can take advantage of this functionality and even if it could, you would still need an always active user session.

As for running TC through the console (>agent.bat start) in an active session, this works for us quite well. We've had no differences running our GUI tests manually or through TeamCity.

ray



[Carel Lotz](#) 78 posts since

Jul 28, 2008 5. **Re: Running Automated GUI tests with TeamCity** Jul 30, 2009 4:56 PM

Running TC through using the console certainly is the way to go. We were also not able to get it working using the service. Unfortunately, for us, this has not proven to be very stable as the tests fall over for half of the time. If we then run the tests on the same machine without using TeamCity they still pass. We use WebAii as testing framework to test our Silverlight application. We have tried different approaches like using a scheduled task on the machine as this has also proven more reliable but this makes it impossible to configure and report on the test-run via TeamCity as you can't control it via the TeamCity portal.

Running Automated GUI tests with...



[Ray Navarette](#) 79 posts since

Dec 7, 2007 **6. Re: Running Automated GUI tests with TeamCity** Jul 30, 2009 4:58 PM

Did you try checking the environment parameters and the memory allocation to the build agent? The testing framework may require a little more memory for overhead than the TC runner is initially configured to use.



[Carel Lotz](#) 78 posts since

Jul 28, 2008 **7. Re: Running Automated GUI tests with TeamCity** Jul 30, 2009 5:05 PM

Interesting - where can I configure this?



[Ray Navarette](#) 79 posts since

Dec 7, 2007 **8. Re: Running Automated GUI tests with TeamCity** Jul 30, 2009 5:13 PM

The environment used by the build agent will be the environment of the user/console it is run under. So, when you open cmd to run the command: 'agent.bat start' the environment of the current cmd session. As for the memory options, take a look inside agent.bat. It looks like you should set the environment variable TEAMCITY_AGENT_MEM_OPTS to configure memory usage. Try something like this first:

```
set TEAMCITY_AGENT_MEM_OPTS=-Xmx256M
```

That will configure the TC jvm with a max heap of 256 MB. I'd think less would be sufficient, but you never know. It's worth a try at least.



[Carel Lotz](#) 78 posts since

Jul 28, 2008 **9. Re: Running Automated GUI tests with TeamCity** Jul 30, 2009 5:15 PM

Thanks, I'll give it a try and see what happens 😊



[Bernard Vander Beken](#) 139 posts since

May 5, 2003 **10. Re: Running Automated GUI tests with TeamCity** Dec 1, 2009 3:34 PM

Carel,

Running Automated GUI tests with...

- Did you manage to make this work?

- Is this still the recommended way to run interactive silverlight tests?

Best regards,

Bernard