

Automated client GUI testing system set up requirements

In order to have the results of running automated tests be stable and predictable, you will need to follow the guidelines below.

These guidelines may evolve as things evolve on new versions of the platforms.

First some more general guidelines :

The jtreg framework runs all the tests and these are run in `othervm` mode for all client tests - as specified in `TEST.ROOT`

```
# Tests that must run in othervm mode
othervm.dirs=java/awt java/beans javax/accessibility javax/imageio javax/sound javax/swing javax/print \
com/apple/laf com/sun/java/accessibility com/sun/java/swing sanity/client demo/jfc \
javax/management sun/awt sun/java2d javax/xml/jaxp/testng/validation java/lang/ProcessHandle
```

If you create a new top-level directory for client tests ie directly under `test/jdk`, then add that directory to the `othervm.dirs` list.

If a test requires access to the desktop - ie creates a window etc such that it is a "headful" task then "@key headful" needs to be added.

"@key printer" should be added to tests that expect a printer

These keywords are defined in the jtreg `TEST.ROOT` file here <https://github.com/openjdk/jdk/blob/f6be7fdf22eede767a0ac29b4f1cb770cfdc0b0f/test/jdk/TEST.ROOT#L14>

There you can see the full list and some additional explanation.

These can then be used when filtering which tests to run. For example

```
make -- test TEST_JOBS=1 TEST="jtreg:jdk_desktop" JTREG='VERBOSE=summary;KEYWORDS=headful&!printer;RETRY_COUNT=2'
```

Will run just the headful tests and exclude headless ones. Or you can do vice versa. `RETRY_COUNT` means that any test that fails should be retried. Sadly this is quite useful as headful tests can sometimes be sensitive to timing etc. These probably indicate a test stability issue but `RETRY` helps you be more sure of that without needing to re-run the entire task. It should also be obvious that you don't touch your keyboard or mouse whilst these are running and that `TEST_JOBS=1` is pointing out that headful tests can't be running concurrently and fighting over input focus etc.

A few of the notes below are more applicable to systems that are used for dedicated testing and need to be stable. For example the one about disabling Windows Auto Update, but the majority are required even for ad hoc testing.

Windows-Specific System set up notes

- Windows defaults to some delays in focus transfer which will cause many test failures. To avoid this you need to do :
`regedit HKEY_CURRENT_USER\Control Panel\Desktop\ForegroundLockTimeout = 0`
- Make sure if the system has JAWS installed, it isn't going to be activated.
- Antivirus software should be configured not do live access checking on the folders containing the JDK being run and the tests.
- Windows Update should be disabled to the extent possible.
- Disable any notifications popups. Windows settings->SystemNotifications & actions Disable everything (especially "Show me the Windows welcome experience screen after updates")
- Show all icons and notifications in taskbar and don't hide them
Right Click on Taskbar Properties (Notification area) Click Customize Enable checkbox "Always show all icons and notifications on the taskbar"

macOS-Specific System set up notes

Disable VoiceOver from System Settings.

- Make the size of the dock as small as possible, otherwise, the dock size(as well as the insets reported to AWT) will be changed over time when the new icons will be adding and deleting from it.

- Disable Bluetooth, Go into System Settings locate the BlueTooth Panel, then "turn bluetooth off"

- Disable the Bluetooth Setup Assistant:

1. Open System Preferences
2. Click Bluetooth
3. Click Advanced...
4. Uncheck Open Bluetooth Setup Assistant at startup if no keyboard is detected
5. Uncheck Open Bluetooth Setup Assistant at startup if no mouse or trackpad is detected
6. Click OK

- Set the sRGB IEC61966-2.1 Profile from the Color Panel of System Setting

- Go into System Settings, locate the Keyboard Panel, and from there select the Shortcuts tab. Disable ALL shortcuts. If you don't it will re-activate Voiceover amongst other problems.

- Go into System Settings, locate the Accessibility Panel, and from there select the General tab. Disable ALL shortcuts. On Voiceover tab - disable Voiceover.

- Go into Security & Privacy => Privacy Tab => Accessibility. Make sure the following apps are allowed to control your computer: *Java* and *Terminal*, that is there's a "tick". Without this, Robot testing will fail (10.15 and later). If you run tests from an IDE, the IDE should also be given this permission.

- Go into System Settings, locate the Keyboard Panel, enable "Use F1,F2,etc keys as standard functional keys"

- Disable any notifications popups. Go into System Settings, locate the Notifications Panel Do not Disturb Turn on From 10.01 AM To 10.00 AM

- Antivirus software should be configured not do live access checking on the folders containing the JDK being run and the tests.

- The tests may require additional permissions on macOS Mojave+, these permissions can be enabled only when the application requests it.

You can preconfigure this by running individual tests which trigger it .. otherwise you'll need to be sitting there when your test job is running.

This seems to be a one time thing. Meaning even if you rebuild "java" macOS remembers but there are no docs I can find which explain this

A list of tests to trigger permission request:

* "Microphone": `jdk.git/test/jdk/javax/sound/sampled/DataLine/DataLine_ArrayIndexOutOfBounds.java`

* "Accessibility": `jdk.git/test/jdk/java/awt/Robot/MouseLocationOnScreen/MouseLocationOnScreen.java`

* "Screen capture": `jdk.git/test/jdk/java/awt/Robot/CheckCommonColors/CheckCommonColors.java`

- Switch off Spotlight indexing (might greatly affect the performance of some tests): `$sudo mdutil -a -i off`

- Disable "ReportCrash" service, it could show modal dialogs for the crashed tests, which break all other tests(see man ReportCrash):

```
$sudo defaults delete com.apple.ProblemReporter
```

```
$sudo defaults write com.apple.CrashReporter DialogType none
```

```
$launchctl unload -w /System/Library/LaunchAgents/com.apple.ReportCrash.plist
```

```
$sudo launchctl unload -w /System/Library/LaunchDaemons/com.apple.ReportCrash.Root.plist
```

- macOS may save the state of application/test windows to the SavedState folder, if a test causes a native crash, immediately after that macOS will show a modal dialog with the text "The last time you opened java it crashed, do you want to reopen it". It is necessary to click reopen otherwise all java tests will hang. To prevent that you need to check these settings:

* Go into System Settings General Enable "Close windows when quitting an application"

* `rm -rf ~/Library/Saved\ Application\ State/com.oracle.java`

This may not be sufficient to ensure that NO windows are re-opened after reboot.

By default macOS has "Reopen windows when logging back in" checkbox enabled in the reboot dialog.

If for some reason (a buggy test for example) some system window such as System Preferences is opened on the desktop, it will stay open and reboot will not help - it will be restored.

By default on macOS "Reopen windows when logging back in" checkbox enabled in the reboot dialog. Reboot the system when changed all the settings above.

If you were to believe <https://support.apple.com/en-us/HT204005> and the section titled "Use the setting in General preferences" then the first piece of advice should have resolved it.

But apparently not. I found that manually clearing the checkbox in the reboot dialog solves it as that will now persist across all reboots.

I have not found an automated way to set this up 😞

A suggestion on the web is to create a shell script that runs on login and deletes the plist file that contains the list of apps to re-open before that file is consulted.

<https://osxdaily.com/2011/08/25/disable-reopen-windows-when-logging-back-in-in-mac-os-x-lion-completely/>

The key pieces of info in there are that the plist file is called `/Users/<my-user>/Library/Preferences/ByHostcom.apple.loginwindow.>LONG UID>.plist`

and the login script (here called `loginfix.sh`) can be activated using

```
defaults write com.apple.loginwindow LoginHook /usr/bin/loginfix.sh
```

Linux-specific system setup notes

- Run using [X.org](#) mode on Ubuntu (or other Linuxes) not the Wayland server otherwise many tests will fail.

On 18.04 X.org is the default. On 21.04 and later it is back to Wayland.

In general configure the pure X server if it is available, if it isn't you are going to see lots of test failures.

On Ubuntu you can either edit `/etc/gdm3/custom.conf` and uncomment `"#WaylandEnable=false"` or on the login screen click the "Gear" Icon and select "Ubuntu on [X.org](#)" or similar.

The "[X.org](#)" is the important part.

- Disable system shortcuts in the system configuration panel.
- Disable any auto-update and any notification update popups: Software&UpdateUpdate tabcheck Never
- Disable color profile: Settings Device Colour Profiles - > uncheck profile if selected

Dock should be placed along the bottom and I recommend to disable auto-hide. Actually this does not seem to be critical but still recommended.

Screen resolution should be at least 1920x1080 and 24 or 32 bit colour. Don't enable desktop fractional scaling or anything like that. It may not cause any failures but is still recommended.