Manual run recording - user

27 January 2015 16:09

Note on 20 April 2015
This method is useful only when non-dev users are to record runs. This is because this method leverages accessibility service which receives very limited information about the widget triggering the event: most (interesting) properties of it are just null. For better, developer-expertise-requiring method, see Manual uia test cases

---

When user manually runs the application, we want to record timestamped logcat dump having "Monitored_API_method_call" logs as well as logs of the most relevant AccessibilityEvents using AccessibilityService [acc].

To further this end, a CatLog app [catlog_app][catlog_github] has to be deployed on a rooted device. Rooted, because READ_LOGS permission is needed [read_logs] [catlog_su]

Furthermore, a standalone Android app (IntelliJ -> new project -> Android -> Gradle: Android Module) having only a class inheriting from AccessibilityService has to be deployed on the device. This class will read the relevant accessibility events and output them to logcat. Such DroidMate app lives in [ui_events_to_logcat].
To make the service work, it has to have appropriate android manifest tags and configuration [acc_setup]. Example of a stub AccessibilityService and configuration can be seen here: [acc_stub]

When the service is installed on the device, user has to go to Settings -> Accessibility and enable the service there. At this point it should start working.

Important note that for the app under monitoring to log API calls, it needs to be inlined and a montitor.apk has to be loaded on the device. To further this end, first run DroidMate at least once on the device - it will install the monitor.apk in appropriate place and keep it there.

Steps for user to record a run: prerequisites
Thus for the user run as described in **Steps to record a run** (below) following prerequisites have to be fulfilled by developer:

- Device has to be rooted // not sure about this, maybe not!
- Device has to have:
  - monitor.apk present in /data/local/tmp
  - Installed inlined .apk subjects
  - Installed UIEventsToLogcat app
  - Installed CatLog app

[catlog_app] https://play.google.com/store/apps/details?id=com.nolanlawson.logcat&hl=en
[catlog_github] https://github.com/nolanlawson/Catlog
[catlog_su] https://github.com/nolanlawson/Catlog/blob/master/Catlog/src/com/nolanlawson/logcat/helper/RuntimeHelper.java
[read_logs] http://stackoverflow.com/questions/11461650/read-logs-permission-on-jelly-bean-api-16

[ui_events_to_logcat] https://hg.st.cs.uni-saarland.de/projects/droidmate-private/repository/changes/dev/ui_events_to_logcat

[acc] http://developer.android.com/reference/android/accessibilityservice/AccessibilityService.html
[acc_setup] http://developer.android.com/guide/topics/ui/accessibility/services.html#manifest
[acc_stub] https://gist.github.com/qihnus/1909616

Steps for user to record a run:

- See prerequisites above
- Enable the UIEventsToLogcat accessibility service in Settings -> Accessibility
- Open CatLog to start log recording.
- Clear logcat.
- Start the subject app
- Click through the app until satisfied.

- After you are done, exit the app (e.g. press "home")
- Disable the UIEventsToLogcat accessibility service in Settings -> Accessibility
- Open CatLog to stop log recording.
- Send the recorded log to person responsible for further processing.

How to process the obtained logcat logs:

- Use your favorite text processing tools.
- Lines with **Monitored_API_method_call** have timestamp and info about given monitored API call.
- Lines with **UIEventsToLogcat** have timestamp and info about given GUI action user made like plain click.