

# Multiplatform GUI testing with fMBT and Python

Antti Kervinen

Intel

[antti.kervinen@intel.com](mailto:antti.kervinen@intel.com)

Test automation day  
Tampere University of Technology, Finland  
June 6<sup>th</sup>, 2014

What would you do to switch on the Bluetooth on your mobile?  
You'd probably...

- pick up your device.
- take a look at the screen.
- act on recognised items
  - tap the Settings icon
- wait for the response
  - wait for the “Bluetooth” text to appear.
- ...

This is a common usage pattern for pretty much every touch UI.

fMBT's GUI Test Interface is a Python API for testing many platforms in this way.

# Multiplatform GUI testing with fMBT and Python

## Python examples

### Android

```
import fmbtandroid
d = fmbtandroid.Device()
d.refreshScreenshot()
d.tapBitmap("settings.png")
d.waitOcrText("Bluetooth")
```

### Tizen mobile

```
import fmbttizen
d = fmbttizen.Device() # pick up a device
d.refreshScreenshot() # take a look at it
d.tapBitmap("settings.png")
d.waitOcrText("Bluetooth")
```

### Chromium OS

```
import chromiumos
d = chromiumos.Device("ssh -p 9222 chronos@localhost")
d.refreshScreenshot()
...
```

### Windows tablet/laptop/desktop

```
import fmbtwindows
d = fmbtwindows.Device("10.123.123.123", password="xxx")
d.refreshScreenshot()
...
```

## GUI Test Interface overview

- Fetching screen contents

- `refreshScreenshot`
  - `refreshView` (Android)

- Convenience methods for taking action on recognised elements

- `tap`
  - `swipe`
  - `verify` × `Bitmap`
  - `wait` `Item`
  - `waitAny` `OcrText`
  - `waitAny` `Text` (Android)

- Recognising elements – count, coordinates, bounding boxes...

- `screenshot().findItemsByBitmap`
  - `screenshot().findItemsByOcrText`
  - `view().findItemsByText` (Android)
  - `view().findItemsById` (Android)

- Executing shell commands

- `shellSOE`

- Visual logging

- `enableVisualLog`

# Multiplatform GUI testing with fMBT and Python

## Visual log contains

- GUI Test Interface calls, parameters, return values, exceptions.
- reference bitmaps, screenshots with recognised items highlighted.

15:55:25.520040 [2\\_switch\\_on\\_bluetooth](#)

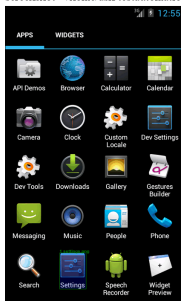
```
15:55:25.520082 tapBitmap('settings.png')
```

```
15:55:25.520242 findItemsByBitmap('settings.png', limit=1)
```



Settings

```
15:55:25.625896 == ['GUIItem("bitmap", bbox=(146, 661, 217, 758), bitmap="/home/ask/testautomationday/settings.png", screenshot="/home/ask/testautomationday/screenshots/20140530-155525-166738-emulator-5554.png")']
```



# Multiplatform GUI testing with fMBT and Python

fmbt-scripter is a tool for

- Writing, editing and recording GUI Test Interface scripts.
- Selecting and updating reference bitmaps on screenshots.
- Executing and debugging scripts.



- fMBT homepage  
<https://01.org/fmbt/>
- fMBT GUI Test Interface wiki  
<https://github.com/01org/fMBT/wiki/GUI-testing>
- fMBT is open source, distributed under LGPL.