



DroidPilot Technical White Paper

An Android Test Automation Solution



Abstract

This paper provides an overview of the prescriptive architecture and the technical approach of how DroidPilot™ as a comprehensive solution to handle Android application testing in an effective and efficient way.

The DroidPilot team

July 2012

The information contained in this document represents the current view of DroidPilot Technology LTD on the issues discussed as of the date of publication. DroidPilot may respond to changing market conditions, it should not be interpreted to be a commitment, and DroidPilot cannot guarantee the accuracy of any information presented after the date of publication.

This White Paper is for informational purposes only. DROIDPILOT MAKES NO WARRANTIES, EXPRESS, IMPLIED OR STATUTORY, AS TO THE INFORMATION IN THIS DOCUMENT.

DroidPilot may have patents, patent applications, trademarks, copyrights, or other intellectual property rights covering subject matter in this document. Except as expressly provided in any written license agreement from DroidPilot, the furnishing of this document does not give you any license to these patents, trademarks, copyrights, or other intellectual property.

Index

1.	Introduction	3
2.	Architecture	3
3.	Working Mode	3
4.	Connection options.....	4
5.	Prerequisites	4
6.	Easy to Use 1-2-3	4
7.	Programming Guide.....	4
8.	Working with DroidPilot	5
9.	Working Scenarios	6
10.	Integrated test management options.....	7
11.	Benefits	7
12.	Solution comparison	8
13.	Need more Info	8

1. Introduction

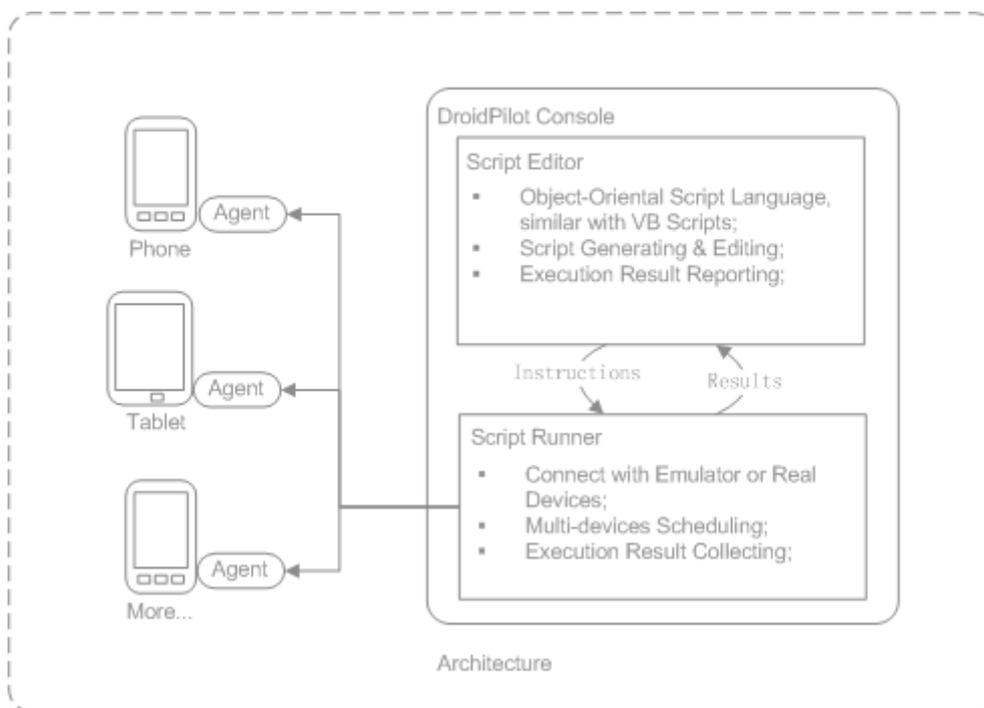
DroidPilot is a test automation tool for Android Native Applications. It can run on both emulator and real devices. The objective of this tool is to perform the automated execution for mobile application testing cases, in order to save the time and effort for manual testing.

2. Architecture

The DroidPilot has two major components:

1. The Agent
2. The Console

The Agent itself is also an Android Native program resides in an Android device. It works with the console and talk to the native applications under test. Users would need to download and install the DroidPilot Agent APK ([Click here to go to the website](#)) to execute respective test scripts on device.



3. Working Mode

DroidPilot support both emulator and device modes. The emulator mode can facilitate script editing and execution without the actual device(s) that host the APK under test. A normal practice is to build up all scripts and well test in the emulator before move up on a real device. The device mode means all executions are conducted by a device attached to the DroidPilot Console.

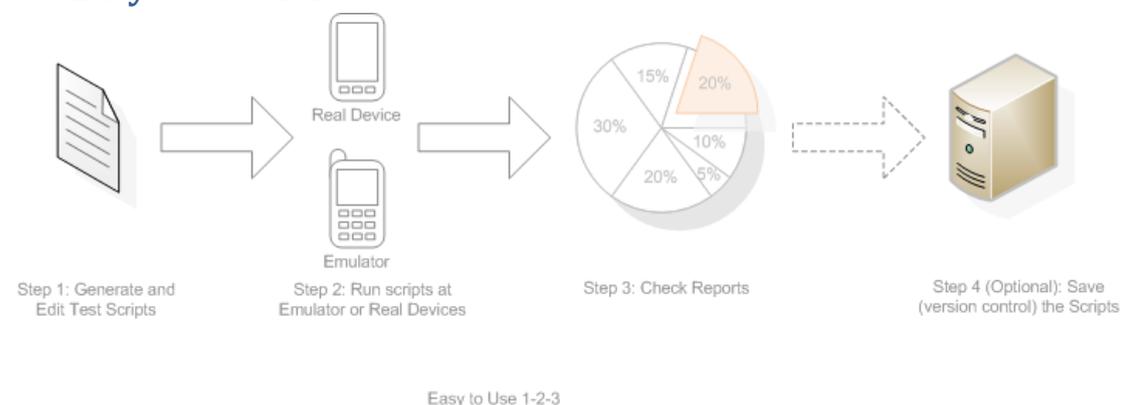
4. Connection options

In the device mode, a device can connect to DroidPilot console by Wi-Fi or USB.

5. Prerequisites

1. The APK under test should be unsigned; otherwise it should add the DroidPilot agent as a trusted process and share in-process data. ([Click here for more info about Trusted Process](#))
2. If the application under test needs to access the device's hardware driver, the device that connects to the DroidPilot Console would have to be rooted. ([Click here for more info about Rooted](#))

6. Easy to Use 1-2-3



Step 1: Generate and edit the scripts by your own;

Step 2: Run the script via emulator or real device;

Step 3: Check reports;

Step 4 (Optional): Save (version control) your scripts at the Test Management Platform - **TestingPilot**;

7. Programming Guide

The following descriptions are the general concepts for the DroidPilot programming guide. For more detailed instructions please refer to <User Manual>.

Preparation

Before the DroidPilot start to work, the following items should be prepared:

1. DroidPilot client side should be installed at local computer;
2. DroidPilot agent should be installed at physical devices or emulator;
3. The Wi-Fi or USB connection between agent and client side should be ready;

Obtain Screen Objects

Before the script editing, the current screen objects of the real devices or virtual devices should be obtained by DroidPilot client side (the console). This is a necessary step for the Keyword Driven programming languages because the script building is base on the obtained objects. Just move to the target screen and click the “Obtain Object” button, all the screen elements will be captured and saved at local location. Users can also see the screen-capture at the Script Editor.

“Drag & Drop” Script Editing

Users can create the auto-test scripts via the table based script editing interface easily. Select an object from the screen-capture, and then select the action (setText, click, checked, etc) from the operation field. If the operation is setText, user can also input the value at the value field.

Grammar

Here are some sample grammars for the script editing:

- If-Else Statement: If ...Then ...[Elseif]...[Else]...End If
- [Object].GetROProperty Method
- Reporter.ReportEvent Method
- [Object].CheckPoint Method
- DB Verification: database(pkgName=#, dbName=#, sql=#, args=#, return=#)
- Sleep(milliseconds) Method
- [Object].Exist Method

Execution

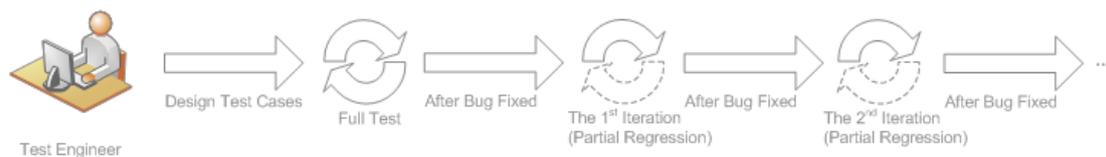
The scripts can be run at both physical device and emulator. User can select the USB or Wi-Fi connection option. Make sure the DroidPilot Agent APK and the APK under test are installed at the device, and they all have been started. Otherwise, DroidPilot Console cannot connect them successfully.

Report

After the script execution, users can check the report to see which steps are passed, which steps are failed, and the detailed failure information.

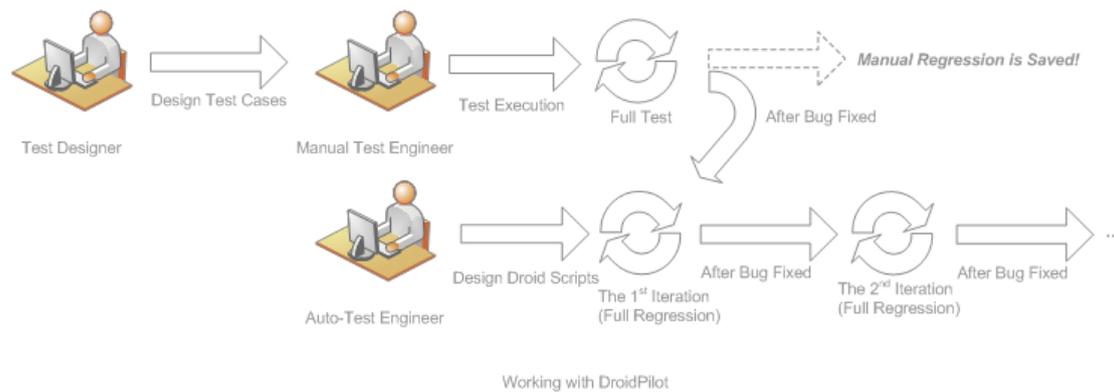
8. Working with DroidPilot

Here describes how the DroidPilot support the auto-test team at the testing life-cycle.



Traditional Working Model

Traditionally, test engineers design test cases for the mobile application which is under test. After that, a full test is performed with some defects found. So that there will be several regression iterations before the final release.



Now, with the automated testing tool -- DroidPilot's support, the regression period can be cut short.

9. Working Scenarios

Scene 1 – Application Developer

Suppose there is an application developer, who is always taking too much time on debugging the program at emulator or at a real device. If there is any feature change, he/she need to re-execute all test cases at the new version again and again at the development phase or at testing phase. To reduce the regression test effort, he/she can involve an auto-test tool. Just build up some auto-test scripts, and take a little effort to maintain them, it can help developer or tester to save a lot of time at the regression test.

Scene 2 - Device Manufacturer

There will be too many models of devices (handsets, tablets, etc) for test engineers to verify the applications on them. That's really a cost. DroidPilot can help manufacturers to solve this problem by the powerful Keyword-Driven testing automation. Different with screen coordinate based test scripts, the Keyword-Driven scripts can once created, run at different screen size devices. Via the multi-device connection interface, the scripts can also run at multiple devices in real time.

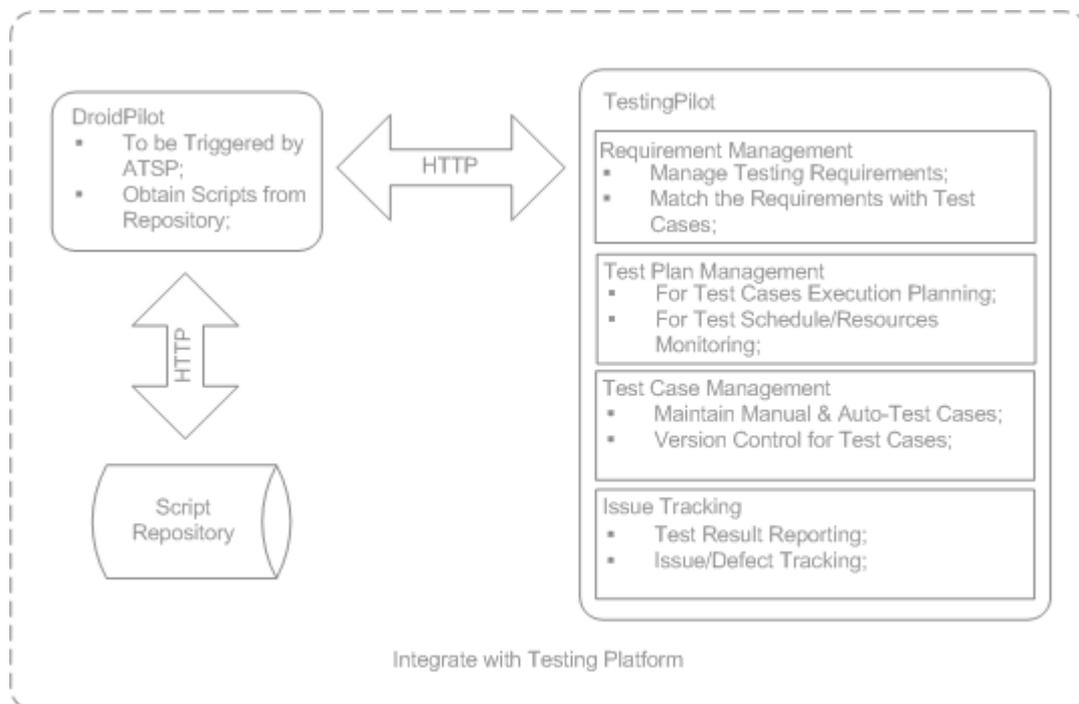
Scene 3 – Application Carrier

This is a most complex case for the consideration of multiple devices from different manufacturers, and the multiple applications from different developers (as an App Store). That's really a heavy task to bind them together to make a compatibility test. Besides the Keyword-Driven Test Scripts, and the Real-time Multi-device Execution, DroidPilot can also integrate with the Test Management Tool. Then it can provide the powerful management capability like Test Requirement Management, Test Plan Management, Test Case Management, Test Scripts' Version Control, Test Execution Scheduling, and Reporting. That might help to perform a reliable quality assurance.

10. Integrated test management options

DroidPilot can be integrated with a powerful testing platform – **TestingPilot**. This platform can provide the following features for DroidPilot:

- Requirement Management;
- Test Plan/Test Case Management;
- Test Case Execution Schedule;
- Version Control for Test Scripts;
- Issue Tracking;



With the supporting by this platform, DroidPilot can make a significant promotion. This integration is optional, user can take DroidPilot as an independent tool.

11. Benefits

1-) Benefit for script Editing;

- Object-Oriental Language Programming; (VB Liked)
- Easy to Use Scripting Interface - Easy & Efficient Script Creation;
- Screen Object Capture;
- Multi-Language Support (Chinese/English);

- Script Recording (coming soon);
 - Cross-Platform Editing (Windows/Linux/Mac);
- 2.) Benefit for Script Running;
- Multi-Channel Connection (USB/Wi-Fi);
 - Multi-Devices in Real Time Execution;
 - Save Time & Effort for Manual Testing;
- 3.) Benefit for Reporting;
- Detailed Reporting;
- 4.) Other Benefits;
- Integrated with Testing Management platform - **TestingPilot**;

12. Solution comparison

	DroidPilot	Cloud Based Tools	Plug-in Based Tools
Support Object Capture?	Yes	Partial	Yes
Support Recording?	Yes*	No	Yes
Support Script Debugging?	Yes	No	Yes
Connect to Emulator & Real Devices?	Yes	Yes	Yes
Connect to multiple devices?	Yes	Yes	Partial
Integrated with Test Management Tools?	Yes	No	Partial

* Available in coming version.

13. Need more Info

Please visit our website (<http://www.droidpilot.com>) for more information, or contact one of our experts (sales@droidpilot.com) to discuss.