Contents

[1. Appium Releases: 2](#_Toc489434580)

[2. Creating a Debug .IPA file from Xcode 2](#_Toc489434581)

[3. iOS Setup for Appium 2](#_Toc489434582)

[4. Setting iOS Real Device with XCUITest : 2](#_Toc489434583)

[5. Third Party tools for Appium . 2](#_Toc489434584)

[5.1. Appium tool identify XCUI Element ( Works only for iOS) 2](#_Toc489434585)

[5.2. Appium Destop 3](#_Toc489434586)

[5.3. Appium Studio 3](#_Toc489434587)

[6. iOS XCUI Capabilities 3](#_Toc489434588)

[7. Appium server capabilities 3](#_Toc489434589)

[8. General Example : 3](#_Toc489434590)

[8.1. Android DesiredCapabilities 3](#_Toc489434591)

[8.2. iOS DesiredCapabilities 4](#_Toc489434592)

[8.3. iOS Swipe to bring Control Centre 4](#_Toc489434593)

[8.4. iOS Touch Action 5](#_Toc489434594)

[8.5. iOS Scrubbing 5](#_Toc489434595)

# Appium Releases:

https://github.com/appium/appium/releases

# Creating a Debug .IPA file from Xcode

(NOTE :- For Android , you can use Release/Debug APK. But for iOS , you need Debug IPA files only)

* Open the project on your Dev machine
* Edit the scheme and click on Archive - change it to Debug
* Click Product → Archive
* Open up the Organizer
* Select your new archive and click export
* Click "Save for Development Deployment"
* (I've been unticking the Rebuild from Bitcode option)

# iOS Setup for Appium

[**https://github.com/appium/appium-xcuitest-driver**](https://github.com/appium/appium-xcuitest-driver)

# Setting iOS Real Device with XCUITest :

Detail Steps about how to use iOS dev cert’s to sign the WebDriverAgent runner

[**https://github.com/appium/appium-xcuitest-driver/blob/master/docs/real-device-config.md**](https://github.com/appium/appium-xcuitest-driver/blob/master/docs/real-device-config.md)

# Third Party tools for Appium .

## Appium tool identify XCUI Element ( Works only for iOS)

After updating my Xcode to latest version and when I try to run my Appium iOS scripts, they were failing .

The reason is , since Appium v1.6.4 , no more support **UIA** class name in Xpath (it’s been deprecated) and you need to change the class name to “**XCUIElementType**”

Example :

**Earlier if you were using Xpath like “UIAApplication[1]/UIAWindow[2]/UIAStaticText[1]”**

**Now the Xpath would look like “XCUIElementTypeApplication[1]/XCUIElementTypeWindow[2]/XCUIElementTypeStaticText[1]”**

Now there are tool to inspect the element built upon XCUI  similar to Appium Inspector.

1. Macaca Inspector :  [https://github.com/chenchaoyi/macaca-appium#run-test-with-xcuitest](https://github.com/chenchaoyi/macaca-appium" \l "run-test-with-xcuitest)
2. Appium-iOS-Inspector : [**https://github.com/mykola-mokhnach/Appium-iOS-Inspector**](https://github.com/mykola-mokhnach/Appium-iOS-Inspector)
3. WebDriverAgent Queries : <https://github.com/facebook/WebDriverAgent/wiki/Queries>

I tried all the three and would recommend the Second one

Steps to use is , just run your appium server and your debug app on device/simulator. Then goto the folder where you have your Appium-iOS-inspector folder and click on the file “iOS Inspector.html”. It will open the browser and you see the screen of your app ( see attached the ScreenShot for REF)

Select any element on the browser and goto “**Absolute XPath**:” you will see something like

“/AppiumAUT/XCUIElementTypeApplication/XCUIElementTypeWindow/XCUIElementTypeOther/XCUIElementTypeOther/XCUIElementTypeOther[7]/XCUIElementTypeOther[3]/XCUIElementTypeOther/XCUIElementTypeButton”

remove the AppiumAUT and your **Xpath** will now looks like

“//XCUIElementTypeApplication/XCUIElementTypeWindow/XCUIElementTypeOther/XCUIElementTypeOther/XCUIElementTypeOther[7]/XCUIElementTypeOther[3]/XCUIElementTypeOther/XCUIElementTypeButton

## Appium Destop

About Appium Desktop (appium version 1.6.4) ( SauceLabs)

Video : <https://saucelabs.com/resources/webinars/an-introduction-to-appium-desktop>

Release : <https://github.com/appium/appium-desktop/releases> ( Install release version)

It’s very easy to setup, start appium server, inspect element on iOS/Android ( when compared to older version of Appium.dmg / Appium CLI)

## Appium Studio

<https://experitest.com/appium-studio/?utm_source=Mailchimp_leads&utm_campaign=appium_studio>

# iOS XCUI Capabilities

[**https://github.com/appium/appium-xcuitest-driver#desired-capabilities**](https://github.com/appium/appium-xcuitest-driver#desired-capabilities)

# Appium server capabilities

<https://github.com/appium/appium/blob/master/docs/en/writing-running-appium/caps.md>

# General Example :

## Android DesiredCapabilities

**public** **void** setUp(**int** port, String deviceId, String OS) **throws** Exception {

ap.startAppium(port);

ap.AppiumURL();

String appiul\_url = ap.AppiumURL();

System.***out***.println("Appium Service Address : - " + appiul\_url);

capa = **new** DesiredCapabilities();

capa.setCapability("appium-version", "1.0");

capa.setCapability("deviceName", deviceId);

capa.setCapability("platformName", "Android");

capa.setCapability("platformVersion", OS);

capa.setCapability("app", "/Users/ramakh01/Desktop/MAP\_Automation/MAPAutomation/Automation/BuildsSMP-AN/SMP-AN-28.4452-dev.apk");

capa.setCapability("platformName", "Android");

capa.setCapability("appPackage", "uk.co.bbc.avtestharnesssmp");

capa.setCapability("appActivity", "uk.co.bbc.avtestharnesssmp.MainActivity");

// capa.setCapability(AndroidMobileCapabilityType.AUTO\_ACCEPT\_ALERTS, true);

**try** {

driver = **new** AndroidDriver<>(**new** URL(appiul\_url), capa);

// capa.setCapability("newCommandTimeout", timeout);

driver.manage().timeouts().implicitlyWait(1000, TimeUnit.***SECONDS***);

} **catch** (Exception e) {

e.printStackTrace();

}

}

## iOS DesiredCapabilities

appiummanager.startAppium(port);

appiummanager.AppiumURL();

String appiul\_url = appiummanager.AppiumURL();

System.out.println("Appium Service Address : - " + appiul\_url);

capabilities = new DesiredCapabilities();

capabilities.setCapability(MobileCapabilityType.APPIUM\_VERSION, "1.6.4");

capabilities.setCapability(MobileCapabilityType.DEVICE\_NAME, dName);

capabilities.setCapability(MobileCapabilityType.UDID, dUDID);

capabilities.setCapability(MobileCapabilityType.PLATFORM\_NAME, "iOS");

capabilities.setCapability(MobileCapabilityType.PLATFORM\_VERSION, dOS);

capabilities.setCapability(MobileCapabilityType.AUTOMATION\_NAME, "XCUITest");//"XCUITest");

capabilities.setCapability(MobileCapabilityType.APP,

"/Users/ramakh01/Desktop/AvTestHarness/iOSApp/AVTestHarness.ipa");

capabilities.setCapability("useNewWDA", true);

capabilities.setCapability("wdaLaunchTimeout", 3000);

capabilities.setCapability("wdaLocalPort", 8100);

## iOS Swipe to bring Control Centre

public void turnWifiON(String TestName, AppiumDriver<WebElement> driver, String networkConnection,

String path, String message, String deviceOS)

throws Exception {

logger = extent.startTest(TestName);

final int height = driver.findElementByClassName("UIAWindow").getSize().getHeight();

final int width = driver.findElementByClassName("UIAWindow").getSize().getWidth();

System.out.println("height"+height);

System.out.println("width"+width);

driver.swipe(width-100, height-5, width-100,0, 50);

driver.findElementByAccessibilityId(networkConnection).click();

Thread.sleep(500);

logger.log(LogStatus.INFO, message + logger.addScreenCapture(capture\_ScreenShot(driver, path, message)));

}

## iOS Touch Action

public void scrolling()

{

// Declare variable that contains the dimensions of the device screen

Dimension winSize;

// Retrieve the actual device dimensions

winSize = iosdriver.manage().window().getSize();

// define two methods that compute the actual coordinates given the percentage

// use the methods to perform the swipe from (20%,62%) to (22%, 35%) and scroll up

int startX = getX(62);

int endX = getX(22);

int startY = getY(20);

int endY = getY(35);

TouchAction touchAction4 = new TouchAction(iosdriver);

touchAction4.press(startX, startY).waitAction(3000).moveTo(endX, endY).release();

iosdriver.performTouchAction(touchAction4);

}

## iOS Scrubbing

public void swipe\_seekbar(WebElement element, AppiumDriver<WebElement> driver, String seekdirection,int seekposition) throws Exception

{

int startX = element.getLocation().getX();

//System.out.println("Startx :" + startX);

// Get end point of seekbar.

int endX = element.getSize().getWidth();

//System.out.println("Endx :" + endX);

// Get vertical location of seekbar.

int yAxis = element.getLocation().getY();

//System.out.println("Yaxis :" + yAxis);

if(seekdirection.equalsIgnoreCase("forward" ))

{

//Thread.sleep(500);

//driver.swipe(endX, yAxis, startX, yAxis, 2000);

driver.swipe(startX, yAxis, endX-seekposition, yAxis, 50);

//driver.swipe(0, 936, 300, 936, 300);

}else if(seekdirection.equalsIgnoreCase("backward"))

{

driver.swipe(endX-seekposition, yAxis, startX+seekposition, yAxis, 50);

}

}