

Borland®

Silk Test 16.0

Testing Mobile
Applications

**Borland Software Corporation
700 King Farm Blvd, Suite 400
Rockville, MD 20850**

Copyright © Micro Focus 2015. All rights reserved. Portions Copyright © 1992-2009 Borland Software Corporation (a Micro Focus company).

MICRO FOCUS, the Micro Focus logo, and Micro Focus product names are trademarks or registered trademarks of Micro Focus IP Development Limited or its subsidiaries or affiliated companies in the United States, United Kingdom, and other countries.

BORLAND, the Borland logo, and Borland product names are trademarks or registered trademarks of Borland Software Corporation or its subsidiaries or affiliated companies in the United States, United Kingdom, and other countries.

All other marks are the property of their respective owners.

2015-02-24

Contents

Testing Mobile Web Applications	4
Testing Mobile Web Applications on Android	4
Testing Mobile Web Applications on a Physical Android device	4
Testing Mobile Web Applications on an Android Emulator	5
Installing a USB Driver	6
Enabling USB-Debugging	6
Manually Setting the Open Agent as a Proxy for an Android Emulator	6
Recommended Settings for Android Devices	7
Configuring the Android Emulator for Silk Test	7
Testing Mobile Web Applications on iOS	9
Testing Mobile Web Applications on a Physical iOS Device	9
Installing the Silk Test Application on an iOS Device	10
Automatically Installing the Silk Test Application on an iOS Device	11
Setting the Proxy for an iOS Device	11
Recommended Settings for iOS Devices	12
Recording Mobile Applications	12
Interacting with a Mobile Device	13
Troubleshooting when Testing Mobile Web Applications	13
Manually Adding a Root Certificate to Test a Secure Web Application	15
Installing the Root Certificate to Test a Secure Web Application	16
Limitations for Testing Mobile Web Applications	16
Clicking on Objects in a Mobile Website	17

Testing Mobile Web Applications

Silk Test enables you to automatically test your mobile applications (apps). Automated testing with Silk Test provides the following benefits:

- It can significantly reduce the testing time of your mobile applications.
- You can create your tests once and then test your mobile applications on a large number of different devices and platforms.
- You can ensure the reliability and performance that is required for enterprise mobile applications.
- It can increase the efficiency of QA team members and mobile application developers.
- Manual testing might not be efficient enough for an agile-focused development environment, given the large number of mobile devices and platforms on which a mobile application needs to function.



Note: Silk Test provides support for testing mobile Web apps and hybrid mobile apps on both Android and iOS devices.

For information on the supported operating system versions and the supported browsers for testing mobile applications, refer to the [Release Notes](#).

Testing Mobile Web Applications on Android

Silk Test enables you to test a mobile application on an Android device or an Android emulator.

Testing Mobile Web Applications on a Physical Android device

To test a mobile application on a physical Android device, perform the following tasks:

1. Connect the device to the machine on which Silk Test is installed.
2. If you are testing this Android device for the first time on this machine, install the appropriate Android USB Driver on the machine.

For additional information, see [Installing a USB Driver](#).

3. Enable USB-debugging on the Android device.
For additional information, see [Enabling USB-Debugging](#).
4. Ensure that the Open Agent is running on the machine to which the Android device is connected.

When testing a mobile Web application, the Open Agent is automatically used as a proxy for the Android device.



Note: A network connection needs to be active on the Android device.

5. If the **Silk Test Web Tunneler** app is not installed on the Android device, Silk Test installs the app to enable the USB connection between the Open Agent and the device.
6. To test a secure mobile Web application over HTTPS, Silk Test copies a root certificate to the device or emulator during hooking. If the certificate is not installed, the **Silk Test Web Tunneler** app displays a message box, stating that the root certificate is not installed. Click on the message box to install the certificate.



Note: If the certificate is not installed automatically during hooking, see [Troubleshooting when Testing Mobile Web Applications](#) or [Manually Adding a Root Certificate to Test a Secure Web Application](#).

7. Close all browsers on the device or emulator, to enable Silk Test to check whether all required certificates for the web application are properly installed and used.
8. Create a Silk Test project for your mobile application.
9. Create a test for your mobile application.
10. Use the **Mobile Recording** feature to record the test against the mobile application.
11. When the **Mobile Recording** feature starts, the **Select Application** dialog box opens. Select the mobile browser that you want to use and start recording.
12. If the selected browser cannot connect to the Web, check if the **Silk Test Web Tunneler** app displays a message stating that the proxy settings are not correct. To manually change the proxy settings:
 - a) Locate the proxy settings of the wireless connection that you are using for the Android device. For additional information on locating the proxy settings, refer to the documentation of your Android device.
 - b) Type `localhost` into the **Proxy** or **Proxy hostname** field.
 - c) Type `9999` into the **Port** field.
 - d) Click **OK**.
13. Replay the test.


An Android device or emulator must not be screen-locked during testing. To keep the device awake while it is connected to a machine, open the settings and tap **Developer Options**. Then check **Stay awake** or **Stay awake while charging**.
14. Analyze the test results.

Testing Mobile Web Applications on an Android Emulator

To test a mobile Web application on an Android emulator, perform the following tasks:


1. Configure the emulator settings for Silk Test.

For additional information, see *Configuring the Android Emulator for Silk Test*.
2. Start the Android emulator.
3. To test a mobile application, set the Open Agent as a proxy for the Android emulator.

 **Note:** Ensure that the Open Agent is running on the machine on which the emulator is installed.

For additional information, see *Manually Setting the Open Agent as a Proxy for an Android Device or Emulator*.
4. To test a secure mobile Web application over HTTPS, install the root certificate of the Web application on the emulator.

For additional information, see *Installing the Root Certificate to Test a Secure Web Application*.

 **Note:** Install the root certificate directly after setting the Open Agent as the proxy, because an issue with the Android emulator will not allow you to install a root certificate when you have otherwise used the Android emulator.
5. Close all browsers on the device or emulator, to enable Silk Test to check whether all required certificates for the web application are properly installed and used.
6. Create a Silk Test project for your mobile application.
7. Create a test for your mobile application.
8. Use the **Mobile Recording** feature to record the test against the mobile application.
9. Replay the test.

An Android device or emulator must not be screen-locked during testing. To keep the device awake while it is connected to a machine, open the settings and tap **Developer Options**. Then check **Stay awake** or **Stay awake while charging**.

10. Analyze the test results.

Installing a USB Driver

To connect an Android device for the first time to your local machine to test your mobile applications, you need to install the appropriate USB driver.

The device manufacturer might provide an EXE with all the necessary drivers for the device. In this case you can just install the EXE on your local machine. If the manufacturer does not provide such an EXE, you can install a single USB driver for the device on the machine.

To install the Android USB driver on Microsoft Windows 7:

1. Find the appropriate driver for your device.

For information on finding and installing a USB driver, see <http://developer.android.com/tools/extras/oem-usb.html>.

2. Connect your Android device to a USB port on your local machine.

3. From your desktop or **Windows Explorer**, right-click **Computer** and select **Manage**.

4. In the left pane, select **Device Manager**.

5. In the right pane, locate and expand **Other device**.

6. Right-click the device name, for example *Nexus S*, and select **Update Driver Software**. The **Hardware Update Wizard** opens.

7. Select **Browse my computer for driver software** and click **Next**.

8. Click **Browse** and locate the USB driver folder.

By default, the Google USB Driver is located in `<sdk>\extras\google\usb_driver\`.

9. Click **Next** to install the driver.

For information on upgrading an existing USB driver or installing a USB driver on another operating system, see <http://developer.android.com/tools/extras/oem-usb.html>.

Enabling USB-Debugging

To communicate with an Android device over the Android Debug Bridge (adb), enable USB debugging on the device.

1. On the Android device, open the settings.

2. Tap **Developer Settings**.

The developer settings are hidden by default. If the developer settings are not included in the settings menu of the device:

a) Depending on whether the device is a phone or a pad, scroll down and tap **About phone** or **About Pad**.

b) Scroll down again and tap **Build Number** seven times.

3. In the **Developer settings** window, check **USB-Debugging**.

4. Set the USB mode of the device to **Media device (MTP)**, which is the default setting.

For additional information, refer to the documentation of the device.

Manually Setting the Open Agent as a Proxy for an Android Emulator

To set the Open Agent as a proxy for your Android emulator, install the Open Agent on the machine from which you want to test the emulator and enable USB debugging on the emulator.

1. Start the Android emulator.
2. On the Android emulator, open the settings.
3. In the **WIRELESS & MORE** section, click **More**.
4. Select **Mobile Networks > Access Point Names**.
5. Select an existing access point to edit it or create a new access point.
6. Type the IP-address of the machine on which the Open Agent is installed into the **Proxy** or **Proxy hostname** field.
7. Click **Port**.
8. Type the port for the Open Agent into the **Port** field. By default, the port number is dynamic, so first you need to set a permanent port number. To change the port number, use the configuration setting `ext.http.proxy.port` in the file `AppData\Roaming\Silk\SilkTest\conf\silkproxy.properties.sample` to set a permanent port number. For example, to set the port number to `9999`, set `ext.http.proxy.port=9999`. Then type the port number into the **Port** field and rename the file `silkproxy.properties.sample` to `silkproxy.properties`.
9. Click **OK**.

The Open Agent is now set as a proxy for your Android device or Android emulator. For additional information on configuring a proxy for your Android device or Android emulator, refer to the documentation of the device or the emulator.



Note: As long as the Open Agent is running, you can use the Internet connection on the mobile device that uses the Open Agent as a proxy. If the Open Agent is not running, the connection will no longer work, and you have to use another connection to connect to the Internet from your mobile device. If you remove the wireless network connection while the device or emulator is still running, the connection to the Open Agent persists until you shut down the device or emulator.

Recommended Settings for Android Devices

To optimize testing with Silk Test, configure the following settings on the Android device that you want to test:

- Enable USB-debugging on the Android device. For additional information, see [Enabling USB-Debugging](#)
- Set a pattern or a PIN to lock the screen of the Android device.
- An Android device must be connected as a media device to the machine on which the Open Agent is running. The USB mode of the Android device must be set to **Media device (MTP)**.
- An Android device or emulator must not be screen-locked during testing. To keep the device awake while it is connected to a machine, open the settings and tap **Developer Options**. Then check **Stay awake** or **Stay awake while charging**.
- To persist your changes for the Android emulator, for example the proxy settings, uncheck the **Wipe user data** check box in the **Launch Options** dialog box of the emulator.

Configuring the Android Emulator for Silk Test

When you want to test mobile applications on an Android emulator with Silk Test, you have to configure the emulator for testing:

1. Install the Android SDK.
For information on how to install and configure the Android SDK, see [Get the Android SDK](#).
2. From Eclipse, click **Window > Android SDK Manager** to start the **Android SDK Manager**.
3. For all Android versions that you want to test with the emulator, expand the version node and check the check box next to **Intel x86 Atom System Image**.
4. Click **Install** to install the selected packages.
5. Expand the **Extras** node and check the check box next to **Intel x86 Emulator Accelerator (HAXM)**.

6. Click **Install** to install the selected packages.
7. Review the *Intel Corporation license agreement*. If you accept the terms, select **Accept** and click **Install**. The **Android SDK Manager** will download the installer to the `extras` directory, under the main SDK directory. Even though the **Android SDK Manager** says `Installed` it actually means that the Intel HAXM executable was downloaded. You will still need to run the installer from the `extras` directory to get it installed.
8. Extract the installer inside the `extras` directory and follow the installation instructions for your platform.
9. In Eclipse, click **Window > Android Virtual Device Manager** to add a new Android Virtual Device (AVD).
10. Select the **Android Virtual Devices** tab.
11. Click **New**.
12. Configure the virtual device according to your requirements.
13. Set the RAM size used by the emulator to an amount that is manageable by your machine.
For example, set the RAM size for the emulator to 512.
14. Set a size for the SD card.

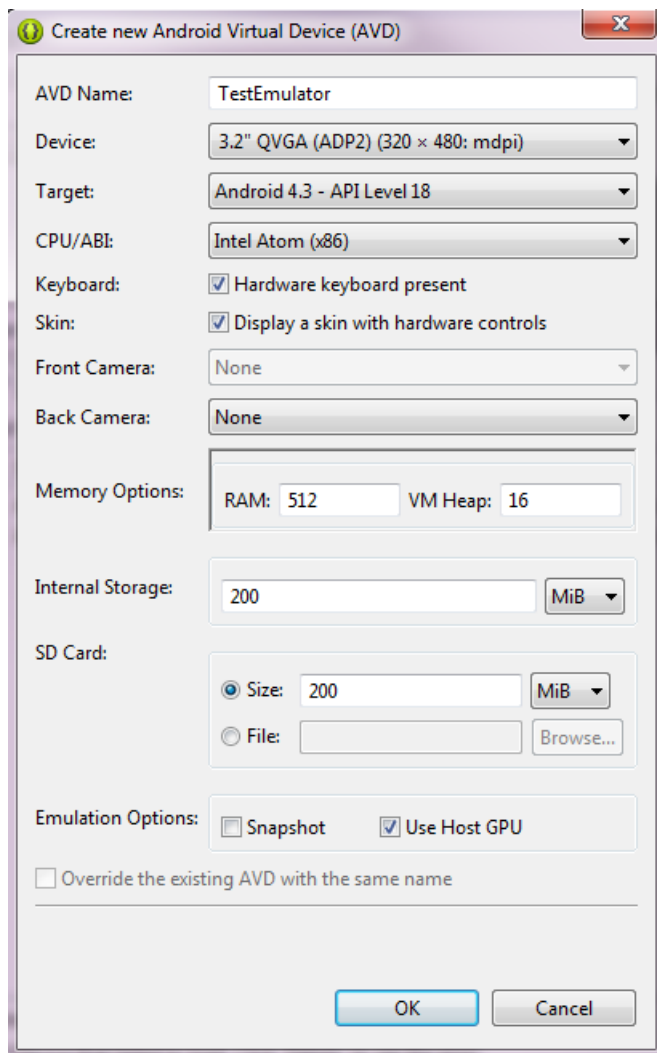


Note: If you do not set a size for the SD card, you need to set the value of the internal storage to 50 MB or more, otherwise you cannot copy the certificate file to the emulator.

15. To enhance the speed of the transactions on the emulator, select the **Intel Atom (x86)** CPU in the **CPU/ABI** field.
16. *Optional:* To enhance the speed of the transactions on the emulator, you can also check the **Use Host GPU** check box in the emulation options.



Note: By setting **Use Host GPU**, you can no longer capture screenshots and would see a black image in the **Mobile Recording** dialog box. However, you could still highlight controls within the **Mobile Recording** dialog box. For additional information, see <https://code.google.com/p/android/issues/detail?id=58724>.



17. Click **OK**.

18. *Optional:* To persist your changes for the Android emulator, for example the proxy settings, uncheck the **Wipe user data** check box in the **Launch Options** dialog box of the emulator.

Testing Mobile Web Applications on iOS

Silk Test enables you to test a mobile application on an iOS device.

Testing Mobile Web Applications on a Physical iOS Device

To test a mobile application (app) on a physical iOS device, perform the following tasks:

1. If you are testing a hybrid app, make the app accessible. For additional information, see [Making Your iOS App Accessible](#).
2. If you are testing a mobile application on an iOS device for the first time on this machine, install iTunes on the machine.
iTunes is required because it contains the device drivers that are needed to test on an iOS device.
3. Install the Silk Test application on the iOS device. For additional information, see [Installing the Silk Test Application on an iOS Device](#).

4. Set `localhost:9999` as a proxy for your iOS device.
For additional information on setting the proxy for an iOS device, see [Setting the Proxy for an iOS Device](#).
5. Connect the device to the machine on which Silk Test is installed.
6. Run a simple test to ensure that the Open Agent is running on the machine to which the iOS device is connected.
7. Open the Silk Test application on the iOS device.
8. To test a secure mobile Web application over HTTPS, install a root certificate for the mobile Web application by using the Silk Test application.
9. Close all browsers on the device or emulator, to enable Silk Test to check whether all required certificates for the web application are properly installed and used.
10. Create a Silk Test project for your mobile application.
11. Create a test for your mobile application.
12. Use the **Mobile Recording** feature to record the test against the mobile application.
13. Replay the test.
The iOS device should not fall into sleep mode during testing. To turn the screen lock and password off, select **Settings > General > Passcode Lock**. In iOS 7, select **Settings > Passcode**.
14. Analyze the test results.

Installing the Silk Test Application on an iOS Device

Install the Silk Test application on an iOS device to enable the USB connection between the Open Agent and the iOS device.



Note: To test an iOS device with Silk Test, the UDID of the iOS device must be registered in the Apple Developer Account of your company.

1. Download Xcode, for example from <https://developer.apple.com/xcode/downloads/>, and install it on a Mac.
The Mac is only required to install the Silk Test application on an iOS device, and does not have to be very fast. For example, a Mac Mini with minimal configuration would be sufficient.
2. Connect the iOS device to the Mac.
3. When a dialog box opens on the iOS device, click **Trust**. You can now use the device in combination with Xcode. After you launch an App for the first time, a Provisioning Profile which matches the developer profile of your company is installed on the device.
4. Copy the archive `SilkTestiOS.zip`, which is located by default under `C:\Program Files (x86)\Silk\SilkTest\ng\iOS` on the Windows machine on which the Open Agent is installed, to the Mac and unpack the archive.



Note: To retrieve the password for unpacking the archive, log in to our [SupportLine site](#) and report an incident with the subject `iOS Password`.

5. Click **File > Open** to import the project to Xcode or click the `.xcodproj` file to open the project.
6. In Xcode, select your device as the target instead of the iOS Simulator, which is set as the target by default.
7. In the project settings, select the developer program of your company.
8. Click on the arrow in the upper left corner or select **Product > Run**.
9. To automatically install the Silk Test application on additional iOS devices used in your company, see [Automatically Installing the Silk Test Application on an iOS Device](#).
10. The Silk Test application on the iOS device is started for the first time.



Note: As soon as the Silk Test application has been successfully started on the iOS device, you can simply tap the icon of the application on the iOS device to start the application.

Automatically Installing the Silk Test Application on an iOS Device

Generate an IPA file and distribute it to enable users in your company to install the Silk Test application automatically on iOS devices.



Note: To test an iOS device with Silk Test, the UDID of the iOS device must be registered in the Apple Developer Account of your company.

1. Download Xcode, for example from <https://developer.apple.com/xcode/downloads/>, and install it on a Mac.
The Mac is only required to install the Silk Test application on an iOS device, and does not have to be very fast. For example, a Mac Mini with minimal configuration would be sufficient.
2. Connect the iOS device to the Mac.
3. When a dialog box opens on the iOS device, click **Trust**. You can now use the device in combination with Xcode. After you launch an App for the first time, a Provisioning Profile which matches the developer profile of your company is installed on the device.
4. In Xcode, compile the Silk Test application.
5. Click **Products > Archive** and generate the IPA file for the Silk Test application.
6. Copy the generated IPA file and a developer disk image for every iOS version that you want to test into the distribution folder that you want to use.
 - a) The developer disk image is located by default in the Xcode installation folder under `xCode/Contents/Developer/Platforms/IOS.platform/DeviceSupport/<iOS_version_number>/`, where *iOS_version_number* is the iOS version of the device that you want to test.
 - b) You need to copy two files for the developer disk image, `DeveloperDiskImage.dmg` and `DeveloperDiskImage.dmg.signature`.
7. On every machine from which you want to test an iOS device, open the folder `%APPDATA%\SilkTest\Conf`.
8. Rename the file `iosApp.properties.sample` to `iosApp.properties`.
9. Open the `iosApp.properties` file and change the file locations to the distribution folder to which you have copied the IPA file and the developer disk image.


When you select an iOS device, with an iOS version for which you have copied a developer disk image, from the **Select Application** dialog, the Silk Test application is installed on the iOS device.

Setting the Proxy for an iOS Device

To set the `localhost` as a proxy for your iOS device, install the Open Agent on the machine from which you want to test the device.

1. On the iOS device, click **Settings > WiFi**.
2. Click on the information button (i) of the active wireless network.
3. In the **Proxy** section, select **Manual**.
4. Type `localhost` into the hostname field.
5. Type `9999` into the port field.

For additional information on configuring a proxy for your iOS device, refer to the documentation of the device.


 **Note:** As long as the Open Agent is running, you can use the Internet connection on the mobile device. If the Open Agent is not running, the connection will no longer work, and you have to use another connection to connect to the Internet from your mobile device. If you remove the wireless network connection while the device is still running, the connection to the Open Agent persists until you shut down the device.

Recommended Settings for iOS Devices

To optimize testing with Silk Test, configure the following settings on the iOS device that you want to test:


- Ensure that the iOS device is running with Xcode and in developer mode.
- To ensure that Apple Safari starts correctly, tap **Settings** > **Safari** and select **Clear Cookies and Data**.
- To make the testing reflect the actions an actual user would perform, disable AutoFill and remembering passwords for Apple Safari. Tap **Settings** > **Safari** > **Passwords & AutoFill** and turn off the **Names and Passwords** setting.
- The iOS device should not fall into sleep mode during testing. To turn the screen lock and password off, select **Settings** > **General** > **Passcode Lock**. In iOS 7, select **Settings** > **Passcode**.

Recording Mobile Applications

 **Note:** Some low-level methods and classes are not supported for mobile Web applications. To be able to correctly replay a test recorded against a mobile Web application, uncheck the **Record native user input** option in the Browser options of Silk Test before recording against the mobile Web application. For additional information, see *Limitations for Testing Mobile Web Applications*.

Once you have established the connection between Silk Test and a mobile device or an emulator, you can record the actions that are performed on a mobile browser on the device to create tests. To record mobile Web applications, Silk Test uses the **Mobile Recording** feature, which provides additional functionality compared to the recorder that is used for standard or Web applications.

The **Mobile Recording** feature displays the screen of the mobile device or Android emulator which you are testing.

 **Note:** If no mobile device is connected to the machine and no emulator is started, the **Mobile Recording** window displays an error message. Connect your mobile device to the machine or start the emulator and then click **Refresh** in the **Mobile Recording** window.

When you perform an action in the **Mobile Recording** feature, the same action is performed on the mobile device.

When you interact with a control on the screen, the **Mobile Recording** feature preselects the default action. A list of all the available actions against the control displays, and you can select the action that you want to perform or simply accept the preselected action by clicking **OK**. You can type values for the parameters of the selected action into the parameter fields. Silk Test automatically validates the parameters.

When you cannot directly interact with a control, for example because other controls are hiding the control, you can click **Toggle Object Hierarchy** in the **Mobile Recording** feature to select the control from the control hierarchy tree.

When you pause the recording, you can perform actions in the screen which are not recorded to bring the device into a state from which you want to continue recording.

When you stop recording, a script is generated with your recorded actions, and you can proceed with replaying the test.

Interacting with a Mobile Device

To interact with a mobile device and to perform an action like a swipe in the application under test:

1. In the **Mobile Recording** window, click **Show Mobile Device Actions**. All the actions that you can perform against the mobile device are listed.
2. Select the action that you want to perform from the list.
3. To record a swipe on an Android device or emulator, move the mouse while clicking the left mouse button.
4. Continue with the recording of your test.

Troubleshooting when Testing Mobile Web Applications

Why does the Select Application dialog box not display my mobile browsers?

Silk Test might not recognize a mobile device or emulator for one of the following reasons:

Reason	Solution
The mobile device is not connected to the local machine.	Connect the mobile device to the local machine.
The emulator is not running.	Start the emulator.
The Android Debug Bridge (adb) does not recognize the mobile device.	To check if the mobile device is recognized by adb: <ol style="list-style-type: none">1. Navigate to <code>C:\Program Files (x86)\Silk\SilkTest\ng\agent\plugins\com.microfocus.silktest.adb_15.0.0.6733\bin</code>.2. Hold Shift and right-click into the File Explorer window.3. Select Open command window here.4. In the command window, type <code>adb devices</code> to get a list of all attached devices.5. If your device is not listed, check if USB-debugging is enabled on the device.
The version of the operating system of the device is not supported by Silk Test.	For information on the supported mobile operating system versions, refer to the Release Notes .
The USB driver for the device is not installed on the local machine.	Install the USB driver for the device on the local machine. For additional information, see <i>Installing a USB Driver</i> .
USB-debugging is not enabled on the device.	Enable USB-debugging on the device. For additional information, see <i>Enabling USB-Debugging</i> .

Why can my mobile device or emulator no longer connect to the Internet?

If you have configured a proxy for every network connection on your mobile device or emulator, and you are currently not recording or replaying any tests, the mobile device or emulator cannot connect to the Internet. For a physical mobile device you can check the connection status in the **Silk Test Web Tunneler** application.

If the mobile device is connected and the Open Agent is running, and the mobile device still cannot connect to the Internet, check if the proxy settings are correct.

To be able to connect to the Internet when the Open Agent is not running, you can temporarily disable the proxy.

Why does Silk Test search for a URL in Chrome for Android instead of navigating to the URL?

Chrome for Android might in some cases interpret typing an URL into the address bar as a search. As a workaround you can manually add a command to your script to navigate to the URL.

Why can I not record on an Android emulator with Android 4.3?

To record on an Android emulator with Android version 4.3, uncheck the **Use Host GPU** check box in the emulator settings.

Why do mobile applications no longer work when I configure the proxy?

Some mobile applications do not use the global proxy that you can set for the WiFi connection. Browsers and some applications like Gmail use the proxy settings, but most other mobile applications ignore the proxy settings and therefore cannot connect to the Internet while the proxy is set.

What do I do if the adb server does not start correctly?

When the Android Debug Bridge (adb) server starts, it binds to local TCP port 5037 and listens for commands sent from adb clients. All adb clients use port 5037 to communicate with the adb server. The adb server locates emulator and device instances by scanning odd-numbered ports in the range 5555 to 5585, which is the range used by emulators and devices. Adb does not allow changing those ports. If you encounter a problem while starting adb, check if one of the ports in this range is already in use by another program.

For additional information, see <http://developer.android.com/tools/help/adb.html>.

Why do I get the error: Failed to allocate memory: 8?

This error displays if you are trying to start up the emulator and the system cannot allocate enough memory. You can try the following:

1. Lower the RAM size in the memory options of the emulator.
2. Lower the RAM size of Intel HAXM. To lower the RAM size, run the `IntelHaxm.exe` again and choose **change**.
3. Open the **Task Manager** and check if there is enough free memory available. If not, try to free up additional memory by closing a few programs.

Why can I not work with a secure website?

If you cannot test a secure website (HTTPS) on a physical mobile device, try the following:

1. Open the **Silk Test Web Tunneler** application on the mobile device to check the following:
 - A certificate is installed for the secure website.
 - The certificate matches the root certificate of the machine on which the Open Agent is installed.

If no certificate is installed or the certificate does not match the root certificate of the machine on which the Open Agent is installed, a yellow warning message is displayed.

2. Click on the warning and select **Ok** to install the certificate. Installing a certificate requires to set a password or a screen lock for the mobile device. If no password or screen lock is set you are prompted to set one during this step.
3. If the certificate is not found on the device the installation fails and an error message displays. Check if the file `root.crt` exists under `sdcard/silk/certs/`.
4. If the file `root.crt` does not exist, copy the file manually by using the **File Explorer**. The certificate might be missing if you have no write permissions on the mobile device.

5. After you have copied the certificate to the device, you can install the certificate by using the **Silk Test Web Tunneler** application or by clicking on the certificate in the file system.

If you cannot test a secure website (HTTPS) on an emulator, manually add the root certificate of the website. For additional information, see *Manually Adding a Root Certificate to Test a Secure Web Application*.

Manually Adding a Root Certificate to Test a Secure Web Application

If you are testing an Android emulator with Android version 4.4 or later, you cannot follow the process described in this topic. For information on how to add a root certificate to test a secure Web application on an Android emulator with Android version 4.4 or later, see [Retrieving the Root Certificate of a Secure Web Application](#).



Note: To perform the steps described in this topic, you must have configured the Open Agent as a proxy for the Android device or Android emulator.

When you are testing a mobile Web application which uses HTTPS on an Android device or Android emulator, each request to open a specific site will automatically generate a certificate for this site on the machine on which the Open Agent is installed. This new certificate is issued to the same domain as the original certificate, replacing the original certificate to enable testing over the SSL connection.

The first certificate that is generated is the root certificate for the mobile Web application.

To be able to test the application with Silk Test, the root certificate needs to be installed on the Android device or Android emulator. By default, the root certificate is copied to the device during hooking. However, if the root certificate is not automatically installed, manually install the root certificate once for each mobile Web application that you want to test.

1. If you are testing a mobile Web application on an Android emulator with Android 4.4 or later, perform the following steps:
 - a) From the Android device or Android emulator, open the mobile Web application that you want to test.
 - b) For example, open www.borland.com.
 - c) Append the following extension to the URL: `/_st_/dynamic/certificate`. For example, the new URL for www.borland.com in the mobile browser is the following: `www.borland.com/_st_/dynamic/certificate`.

2. Open the mobile Web application that you want to test. If it is the first time that you open the mobile Web application, the Open Agent generates the modified root certificate for the application.
3. On the machine on which the Open Agent is installed, go to the folder where the root certificate is located.

By default, this is the folder `%Appdata%\Silk\SilkTest\certs\authority`.

4. Copy the root certificate file `root.crt`.
5. Paste the root certificate file to the root folder in the storage of your Android device.

If you are testing on an Android emulator, the Open Agent automatically copies the certificate to the root directory of the emulator.



Note: To enable the Open Agent to copy the certificate to the emulator, configure a size for the SD card in the emulator settings.


6. If you are testing on a physical Android device, install the certificate from the storage into your Android device.


For additional information about how to install a certificate from the storage, refer to the documentation of your Android device or Android emulator.

7. If you are testing on an Android emulator:
 - a) Navigate to **Settings > Security > Install from SD card** on the emulator.

- b) Click **OK** to install the certificate.
 - c) *Optional*: Navigate to **Settings > Security > Trusted credentials > USER** to verify that the certificate is installed on the emulator.
8. Close all browsers on the device or emulator, to enable Silk Test to check whether all required certificates for the web application are properly installed and used.

Installing the Root Certificate to Test a Secure Web Application

 **Note:** If you are testing a physical Android device, or an Android emulator with an Android version prior to 4.4, see [Manually Adding a Root Certificate to Test a Secure Web Application](#).

 **Note:** To perform the steps described in this topic, you must have configured the Open Agent as a proxy for the Android device or Android emulator.

When you are testing a mobile Web application which uses HTTPS on an Android device or Android emulator, each request to open a specific site will automatically generate a certificate for this site on the machine on which the Open Agent is installed. This new certificate is issued to the same domain as the original certificate, replacing the original certificate to enable testing over the SSL connection.

The first certificate that is generated is the root certificate for the mobile Web application.

To be able to test the application with Silk Test, the root certificate needs to be installed on the Android device or Android emulator. By default, the root certificate is copied to the device during hooking. However, if the root certificate is not automatically installed, manually install the root certificate once for each mobile Web application that you want to test.

1. From the Android emulator, open the mobile Web application that you want to test.
For example, open www.borland.com.
2. Append `/_st_/dynamic/certificate` to the URL and go to the new URL.
For example, the URL for www.borland.com in the mobile browser is the following: `www.borland.com/_st_/dynamic/certificate`.
3. Type a name for the certificate into the **Certificate name** field in the certificate download dialog box.
4. Leave the default setting, **VPN and apps**, in the **Credential use** list box.
5. Click **OK**. The certificate is installed on the emulator.
6. Close all browsers on the device or emulator, to enable Silk Test to check whether all required certificates for the web application are properly installed and used.

Limitations for Testing Mobile Web Applications

The support for playing back tests and recording locators on mobile browsers is not as complete as the support for the other supported browsers. The following list lists the known limitations for playing back tests and recording locators on mobile browsers:

- The following classes, interfaces, methods, and properties are currently not supported for mobile Web applications:
 - `BrowserApplication` class.
 - `CloseOtherTabs` method
 - `CloseTab` method
 - `ExistsTab` method
 - `GetActiveTab` method
 - `GetSelectedTab` method

- `GetSelectedTabIndex` method
- `GetSelectedTabName` method
- `GetTabCount` method
- `OpenTab` method
- `SelectTab` method
- `DomElement` class.
 - `DomDoubleClick` method
 - `DomMouseMove` method
 - `GetDomAttributeList` method
- `DomForm` class. All methods and properties in this class are not supported for mobile Web applications.
- `DomRadioButton` class.
 - `RadioListItemCount` property
 - `RadioListItems` property
 - `RadioListSelectedIndex` property
 - `RadioListSelectedItem` property
- `DomTable` class. All methods and properties in this class are not supported for mobile Web applications.
- `DomTableRow` class. All methods and properties in this class are not supported for mobile Web applications.
- `IClickable` interface.
 - `Click` method. You can use clicks on Web applications running on an Android device, but not on an iOS device.
 - `DoubleClick` method
 - `PressMouse` method
 - `ReleaseMouse` method
- `IKeyable` interface. All methods and properties in this interface are not supported for mobile Web applications.
- Image recognition is not supported for iOS. When you are testing a Web application on an iOS device, you can only use image verifications.
- XPath logical operators are supported only for standard HTML attributes, and are not supported for properties and custom Silk Test attributes. For example, the logical operators are not supported for the `textContent` attribute and the `innerText` attribute. Expressions built with these operators are always case-sensitive, independent of the Silk Test setting.
- XPath logical operators are not supported on stock Android browser on Android versions prior to version 4.4.
- Recording in landscape mode is not supported for emulators that include virtual buttons in the system bar. Such emulators do not correctly detect rotation and render the system bar in landscape mode to the right of the screen, instead of the lower part of the screen. However, you can record against such an emulator in portrait mode.

Clicking on Objects in a Mobile Website

When clicking on an object during the recording and replay of an automated test, a mobile website presents the following challenges in comparison to a desktop website:

- Varying zoom factors and device pixel ratios.
- Varying screen sizes for different mobile devices.
- Varying font and graphic sizes between mobile devices, usually smaller in comparison to a website in a desktop browser.

- Varying pixel size and resolution for different mobile devices.

Silk Test enables you to surpass these challenges and to click the appropriate object on a mobile website.

When recording a test on a mobile device, Silk Test does not record coordinates when recording a `Click`. However, for cross-browser testing, coordinates are allowed during replay. You can also manually add coordinates to a `Click`. Silk Test interprets these coordinates as the HTML coordinates of the object. To click on the appropriate object inside the `BrowserWindow`, during the replay of a test on a mobile device, Silk Test applies the current zoom factor to the HTML coordinates of the object. The device pixel coordinates are the HTML coordinates of the object, multiplied with the current zoom factor.

If the object is not visible in the currently displayed section of the mobile website, Silk Test scrolls to the appropriate location in the website.

Example

The following code shows how you can test a `DomButton` with a fixed size of 100 x 20 px in your HTML page.

During replay on a different mobile device or with a different zoom factor, the `DomButton` might for example have an actual width of 10px on the device screen. Silk Test clicks in the middle of the element when using the code above, independent of the current zoom factor, because Silk Test interprets the coordinates as HTML coordinates and applies the current zoom factor.

Index

A

- adding root certificates
 - Android 15
 - Android emulators 16
- Android
 - configuring emulator 7
 - enabling USB-debugging 6
 - installing USB drivers 6
 - prerequisites 15
 - recommended settings 7
 - setting proxy for emulator 6
 - testing 4
 - testing on emulators 5
 - testing on physical devices 4
 - troubleshooting 13
- Android emulators
 - prerequisites 16

C

- Click
 - mobile Web 17

D

- device not connected
 - mobile 13

I

- installing USB drivers
 - Android 6
- iOS
 - installing Silk Test application 10
 - installing Silk Test application automatically 11
 - recommended settings 12
 - setting proxy 11
 - testing 9
 - testing on physical devices 9

M

- mobile

- troubleshooting 13
- mobile applications
 - recording 12
 - testing 4
- mobile browsers
 - limitations 16
- mobile devices
 - interacting with 13
 - performing actions against 13
- mobile recording
 - about 12
- mobile testing
 - Android 4
 - Android emulators 5
 - iOS 9
 - overview 4
 - physical Android devices 4
 - physical iOS devices 9
- mobile Web
 - Click 17
- mobile Web applications
 - limitations 16

P

- proxy server
 - setting for Android emulator 6
 - setting for iOS 11

R

- recording
 - mobile applications 12
- root certificates
 - adding 15
 - adding, Android emulators 16
 - generating 15
 - generating, Android emulators 16

T

- troubleshooting
 - mobile 13