Define capabilities

The following tables provide the capabilities supported by the Perfecto Lab Automation for Selenium and Appium.

For more information, see Supported Appium capabilities.

On this page:

Perfecto Lab credentials

Capability Name	Meaning	Values /Examples
securityToken	User's personal security token (recommended to use as authentication).	
user	**Deprecated**. The name of the user running the operation. **Note:** Going forward, all authentication requires a security token. For information on obtaining a security token, see Generate security tokens.	
password	**Deprecated**.The password for the user. **Note:** Going forward, all authentication requires a security token. For information on obtaining a security token, see Generate security tokens.	

Important: Use the ***securityToken*** for authentication. The user/password combination is no longer supported for authentication.

Select a mobile device

Define the capabilities to select a device in the Continuous Quality Lab according to device attributes (instead of a specific device ID) and define the CQ Lab credentials.

Note: You can still select a specific device using the **deviceName** capability.

Best Practice: Use auto-generated capabilities as the basis for selecting either Mobile or Web devices.

Capability Name	Meaning	Values / Examples
deviceName	The deviceId	For example: 345304573489573498
deviceType	The platform type	Web, Mobile
description	The device description	
location	The device location	For Example: NA-US-BOS
manufacturer	manufacturer	For example: Apple, Samsung, HTC, Microsoft
model	The device model.	For example: iPhone-5S, Galaxy S III, Xperia Z, 9100, HTC One
network	The device network	For example: AT&T, Verizon
openDeviceTimeout	The timeout, in minutes, to wait for a specific device in case it is not available at the start of the script (use with caution)	Max: 15 minutes
platformName	The device operating system	For example: Android, Windows, iOS

platformVersion	The device operating system version	For example: 9.3.1
resolution	The resolution of the device screen	For example: 1920x1080
screenshotFormat	Type of image	**.jpg** (default), .png, .bmp
automationInfrastructure	Device automation infrastructure of the mobile	For example: XCUITest, UIAutomation (for iOS), UIAutomator1, UIAutomator2 (for Android)
audioPlayback	Determine if device playback audio should be added to the video recording of the execution or not	true, false. Default is false

Example

```
DesiredCapabilities capabilities = new DesiredCapabilities("mobileChrome", "", Platform.ANY);
String host = "mymobilecloud.perfectomobile.com";
capabilities.setCapability("platformName", "Android");
capabilities.setCapability("platformVersion", "4.4");
capabilities.setCapability("user", "myUser");
capabilities.setCapability("securityToken", "myToken");
URL url = new URL("https://" + host + "/nexperience/perfectomobile/wd/hub");
RemoteWebDriver driver = new RemoteWebDriver(url, capabilities);
```

Auto-selection of leading devices

The limitation in using a specific device is that sometimes the device may be busy running a different script or may be disabled. The better option for scripts running automatically is to supply the necessary device characteristics and let Perfecto automatically select the device from the available devices. When the test script does not define a specific device, Perfecto selects a leading device for testing. This is the default configuration in public cloud instances. For private clouds, Perfecto Support can configure leading devices upon request.

The leading devices feature ensures that your tests always run again the most relevant, stable, popular devices with the highest possible OS version. If a leading device is not available, Perfecto selects the device with the highest OS version instead. In particular, Perfecto selects devices based on the following guidelines:

1. If **`model`** or **`manufacturer`** are selected, but no `platformVersion` is selected, Perfecto sets the **`platformVersion`** to **`latest`**.
2. If **`model`** and **`manufacturer`** are not selected and the Leading Device feature is enabled, Perfecto selects one of the leading devices with the highest OS version available.
3. In either case, if the **`platformVersion`** is not selected, Perfecto selects the highest available OS version. Prior to this enhancement, the OS selection was random.
4. If **`model`** is set to **`leading`**, Perfecto selects a random leading device with the highest OS version available. If a leading device is not available, the allocation fails.
5. If **`model`** is not selected, Perfecto selects a random leading device with the highest OS version available. If a leading device is not available, Perfecto selects a random device with the highest OS version available.

If you are an automation engineer, this means that when you use the `Select device` command to select a device based on attributes, you can:

- Set the `model` capability to `leading` to have the script test one of the leading devices. For example:

  ```
  capabilities.setCapability(”model", ”leading”);
  ```

- Set the `platformVersion` capability to `Latest` to make sure the script tests the latest OS version. For example:

  ```
  capabilities.setCapability("platformVersion", "Latest”);
  ```

The following table lists examples.

Capabilities specified	Situation	Result
`platformName=Android`	A leading Android device is available.	A random leading Android device is selected.
`platformName=Android`	A leading Android device is **NOT** available.	A random Android device is selected.
`platformName=Android`, `model=leading`	A leading Android device is available.	A random leading Android device is selected.
platformName=android, `model=leading`	A leading Android device is **NOT** available.	Allocation fails.
`model=Samsung Galaxy S10`	Samsung Galaxy S10 is available.	The specific device is allocated.

model=Samsung Galaxy S10	Samsung Galaxy S10 is NOT available.	Allocation fails.
`model`=Samsung galaxy S10, `platformName`=iOS	Samsung Galaxy S10 is available.	Allocation fails.
`model`=Samsung galaxy S10, `platformName`=UK	There is a Samsung Galaxy S10 in Boston, but not in the UK.	Allocation fails.

For the public cloud, the following devices are currently defined as leading devices.

Android	iOS
Galaxy S10	iPhone 8
Galaxy S10+	iPhone 8 Plus
Galaxy S10e	iPhone X
Galaxy S9	iPhone XS
Galaxy S9+	iPhone XS Max
Galaxy S8	iPhone XR
Galaxy S8 Plus	
Pixel 3	
Pixel 3 XL	

Mobile browser

Capability Name	Meaning	Values
browserName	Browser application name	chrome, mobileChrome, safari, mobileSafari, mobileOS, mobileDefault
version	Browser version	for example: 53, 49
platform		Any

Regular expressions

Regular expressions including wildcards can be used. The capability values are ***case sensitive***.

- OR - e.g. "ATaT|T-Mob" for network means either ATaT (for AT&T) or a network beginning with 'T-Mob'.
- NOT - e.g. "(?!(Amazon|Apple)).*" for manufacturer means neither Amazon nor Apple.
- contains - e.g. ".*Galaxy.* for model means the model contains the string 'Galaxy' and would match any of {Galaxy S6, Galaxy S9, Galaxy 8+}.

Capabilities for development

Capability Name	Meaning
eclipseExecutionId	The Eclipse execution id
scriptName	Name used as the Report Name for the Report Library and Live Stream interfaces (see also the ***Smart Reporting capabilities*** below)

Click *here* for a breakdown of supported Appium capabilities.

Desktop Web testing

Web VM configuration capabilities

Set the capabilities to define the VM to be used for your Web app testing.

Capability Name	Meaning	Values	Comments
platformName	The VM OS	Windows, Mac	
platformVersion	The OS version	See lists of supported versions	
browserName	The browser running on VM	Internet Explorer, Chrome, Firefox, Edge, Safari	
browserVersion	The browser version.	See here for supported browser versions. See Note1 for special values that may be used.	
resolution	The VM display resolution	1024x768 (default) See here for complete list of supported resolutions	
location	Location of Perfecto Web machine facility, when using a virtual web platform. When accessing *physical Mac devices* use same values as for Mobile selection (See Note2 below)	US East, EU Germany, AP Sydney	
deviceType	The platform type	Web, Mobile	
addHostsRecord	Add custom records to the VM's *hosts* file. (See Note3 below.)	<key, value> entry, • key is the *IP-address* • value is *DNS hostname* to add to the file	
seleniumVersion	Version of the Selenium server to configure	3.0.1, 3.4.0, 3.8.1	**Default:** 3.8.1
chromedriverVersion	Version of the Selenium driver classes for the Chrome browser.	2.73, 2.74, 2.75, 2.76, 2.78, 2.79, 2.80, 2.81, 2.83, 2.84	**Default:** The number after the decimal point should align with the browser version. For example, for Chrome 73, use driver 2.73; for Chrome 74, use driver 2.74.
geckodriverVersion	Version of the Selenium driver classes for the Firefox browser.	0.16.1, 0.17.0, 0.18.0, 0.19.0, 0.19.1, 0.20.0, 0.20.1, 0.21.0, 0.22.0, 0.23.0, 0.24.0, 0.25.0, 0.26.0	**Default:** 0.26.0
iedriverVersion	Version of the Selenium driver classes for the IE browser.	3.0, 3.1, 3.2, 3.3, 3.4, 3.5, 3.6, 3.7, 3.8, 3.9, 3.10, 3.11, 3.13, 3.14, 3.141, 3.141.59	**Default:** 3.141.59

Note 1: The **browserVersion** capability supports the following values in addition to a version number:

- ***latest***: This will always run the latest supported version for the selected browser.
- ***latest-1***: This will run the version previous to the latest version supported.
- ***latest-2***: This will run a version two previous to the latest version supported.
- ***beta***: This will always run the latest beta-testing version for the selected browser.

These values will be translated to the appropriate version number and will be listed in the execution report with the version number.

Note 2: The **location** capability is optional when **platformName** capability is ***Windows*** but mandatory when **platformName** is ***Mac***.

Example

```
DesiredCapabilities capabilities = new DesiredCapabilities("mobileChrome", "", Platform.ANY);
String host = "mymobilecloud.perfectomobile.com";
capabilities.setCapability("platformName", "Windows");
capabilities.setCapability("platformVersion", "10");
capabilities.setCapability("browserName", "Chrome");
capabilities.setCapability("browserVersion", "latest");
capabilities.setCapability("resolution", "1366x768");
capabilities.setCapability("location", "US East");
capabilities.setCapability("chromedriverVersion", "2.23");
capabilities.setCapability("seleniumVersion", "3.8.1");
URL url = new URL("https://" + host + "/nexperience/perfectomobile/wd/hub/fast");
RemoteWebDriver driver = new RemoteWebDriver(url, capabilities);
```

Note 3: Set up the records to add to the *hosts* file as a dictionary prior to setting the capability value. The following examples would generate the following lines in the VM's *hosts* file:

```
...
# localhost name resolution is handled within DNS itself.
#    127.0.0.1       localhost
#    ::1             localhost
ip1   url1
ip2   url2
```

Java example

```
Map<String, String> hostsRecords = new HashMap<>();
hostsRecords.put("ip1", "url1");
hostsRecords.put("ip2", "url2");

capabilities.setCapability("addHostsRecord", hostsRecords);
```

JavaScript example

```
var hostsRecords={};
hostsRecords ["ip1"]="host1";
hostsRecords ["ip2"]="host2";

...
var capabilities = {
    ...
    'addHostsRecord': hostsRecords,
    ...
}
```

Web application capabilities

Capability Name	Meaning	Values
takesScreenshot	System takes screenshots of application at different points during execution and attaches them to the execution report.	true | false (default: **false**)
screenshotOnError	System takes screenshot of application at point where an error status is reported. Screenshot is attached to the execution report	true | false (default: **true**)

Notes: Taking screenshots affects the execution time of the Desktop-Web test. Screenshot capabilities support Selenium commands (not, for example, Visual Analysis or Assert).

Example

```
capabilities.setCapability("takesScreenshot", true);
capabilities.setCapability("screenshotOnError", false);
```

Smart Reporting capabilities

Test analysis with Smart Reporting uses different test identifying parameters as flags associated with the test reports. These identifying items are used to filter, select, or just to easily identify the test report. These values may be associated to the test using the reporting SDK or alternatively using the following DesiredCapability fields:

Capability Name	Meaning	Values
report.projectName	Identifier of the test run project name	String
report.projectVersion	Version number associated with the project	String
report.jobName	CI Job name for this test	String
report.jobNumber	CI Job number for this test	Integer
report.jobBranch	Name of test branch, if relevant	String
report.tags	Any tags user associated with this test	Strings, separated by a comma (',')
report.customFields	Any set of custom parameters to associate with the test run	Strings in the format "**key=value,key=value**..."

Any values for the *project* or *job* information will be overwritten by settings using the SDK. Values provided for the *tags* and *customFields* will be merged, with priority given to values set by the SDK. See also *scriptName* capability above.

Example:

```
String browserName = "mobileOS";
DesiredCapabilities capabilities = new DesiredCapabilities(browserName, "", Platform.ANY );
String host = perfectoLabURL;
capabilities.setCapability("securityToken", myToken);
capabilities.setCapability("report.projectName", "test_fail_to_open");
capabilities.setCapability("report.projectVersion", "2.0");
capabilities.setCapability("report.tags", "spring,forward,tag1");
capabilities.setCapability("report.jobName", "myTestJob");
capabilities.setCapability("report.jobNumber", 8);
capabilities.setCapability("report.jobBranch", "master");
capabilities.setCapability("report.customFields", "cc=custom,mm=fewer");
```

Perfecto report and video

Faster test execution performance with the option to specify whether a report or video will be created during script execution. This means more tests can be run per hour.

Capability Name	Meaning	Values
outputVideo	The test execution output video	
outputVisibility	The report access user permissions	private, group, public

Example

```
capabilities.setCapability("outputVideo", false);
capabilities.setCapability("outputVisibility", "public");
```

Wind Tunnel

Capability Name	Meaning
windTunnelPersona	The name of a pre-defined persona
windTunnelPersonaKey	The repository key of a defined persona json file
windTunnelLocation	The device location, as coordinates *
windTunnelLocationAddress	The device location, as an address *
windTunnelOrientation	The device orientation

windTunnelVNetwork	The virtual network profile
windTunnelBackgroundRunningApps	The applications to run in the background

* This is the device location to simulate. It is different from the *location* capability, used for selecting a mobile device, that identifies the actual device location.

Supported Appium capabilities

Learn more on Appium specific supported capabilities here.

Also in this section:

- Use capabilities to select a device
- Select a device based on supported features