Single test report (STR)

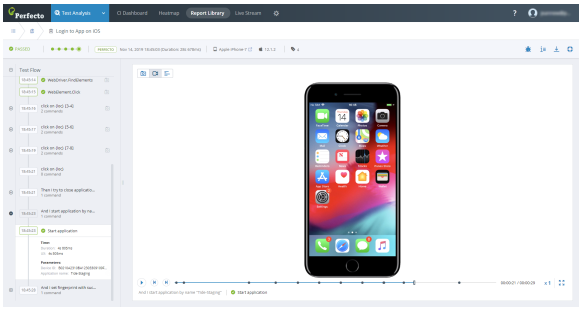
The **Test Analysis** views provide a good overview of the trends and highlights of your test programming. To get more specific information about a particular test, drill down to the Single Test Report (STR) by clicking test report in the **Report Library**.

On this page:

Overview

Clicking the name of a test report in the **Report Library** opens the specific Single Test Report (STR), as shown in the following figure. The report displays a list of the logical test steps on the left and a visual area for screenshots, video, and text artifacts, if available, on the right. The visual area includes a video timeline at the bottom, with the timeline points corresponding to the logical steps on the left.

Note: Depending on the testing framework you use to run your tests, not all features described in this section may apply. For example, if you work with XCUITest or Espresso, you do not see individual test steps, you cannot add custom failure reasons from within tests, and the report may only include a single screenshot. For details on reporting limitations, see the dedicated framework documentation in the Automation testing section.



Logical steps

The left panel of the STR view shows a list of the logical steps that comprise the test (as named in the *testStep()* method).

Reports for native automation executions that activate nested scripts include the steps of both the main script and the nested script. The commands of the nested scripts are identified by a special symbol ("**</>**"). Clicking a logical step reveals a view of the artifacts (video, screenshots, expected vs. actual values) associated with the particular command/step.

Command information

You can click a command to display detailed information about the command execution, including:

- **Timer information:** Displays the Perfecto Timer and UX Timer values when the command was executed.
- **Parameter information:** Identifies the following:
  - The device used for the command
  - The UI element (if the command accessed a UI element)
  - The text sent to the UI element (if the command inserted text)

    Note: If the text was sent as a Secured String, the text value does not display. Instead, it appears as: "*******"
  - Other information, such as parameters for visual analysis, assertion information, or UI element attribute values

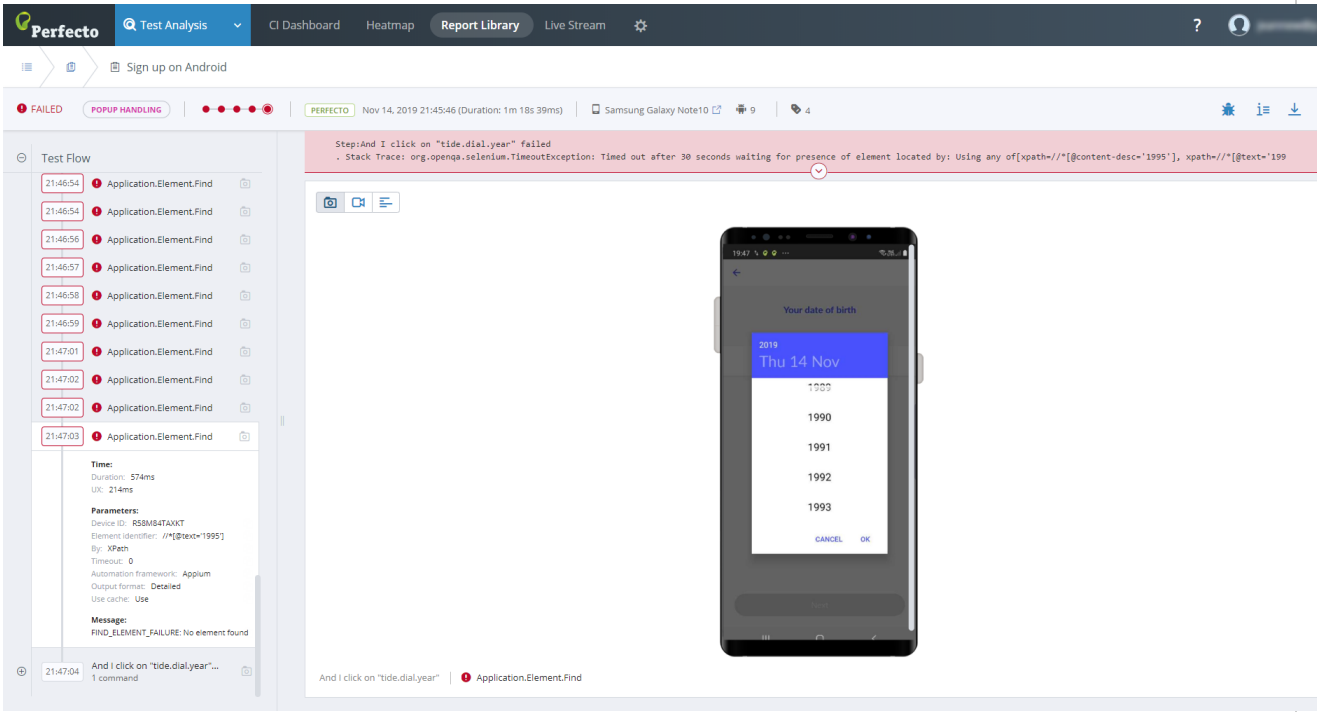
Visual artifact area

The right panel of the STR view presents visual artifacts, such as screenshots or videos from the test run.

When you view the video, the timeline includes indicators that highlight the times at which the logical steps occurred. Moving the pointer over any of these points displays a tooltip that identifies the corresponding logical step.

Error messages

When the test script displayed in the STR generated an error or failure message, the message is displayed at the top of the visual artifact area. At first, only the header line of the error message is displayed on a red background. To see the full message, together with a stack dump (if relevant), click the down arrow below the error message:



STR header area

The Single Test Report header consists of two lines that include the following information:

- First line:
  - The Report Library button . Click to go back to the **Report Library** view, regardless of any navigation to other test report views.
  - The name of the current test.
  - If the current test is part of a scheduled retry, the retry navigation bar: 2 out of 2 retries ‹ Prev Next ›

    Click **Prev** or **Next** to scroll through the individual retries.
- Second line:
  - The status of the test run of this STR (**Passed**, **Failed**, **Blocked**, **Unknown**).
  - The history graph. It shows five runs, similar to the history graph in the **Report Library** view. Clicking a node in the graph navigates to the STR of the selected run.

> The History mechanism defines test similarity by test name and execution capabilities. If there is a difference in either the name or the capabilities (for example, the **`osVersion`** capability is included in one test run but not in another), the tests are considered 'not similar'. As a result, they are not connected in history.

The test run whose details are described in this report is displayed as a double-ring in the history (). This makes it easier to identify when this test run was executed relative to other test runs. When reading the history graph, keep in mind that:

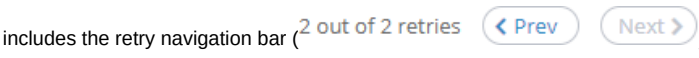
- The latest test run is always represented by the right-most node.
- No more than five nodes appear in the history graph. If the specific run occurred prior to the five latest runs, the graph shows a break represented by three dots ().
- The color of the node represents the test result status for that particular run, where green means 'passed', yellow means 'blocked, and red means 'failed' ().
- Move the pointer over a node to display a tooltip with details for that run.
- Arrows around an icon () identify test runs with scheduled retries that have been collapsed into a single test report.

- **Run information** - Start time and duration information of the test's run.
- **Device information** - information on the device or devices used for the test run.
  - **Activate interactive session** icon - Opens the device in a Perfecto Lab interactive session. If the device is not available, the Perfecto Lab will notify the user to select another device.
- **Tags** - list of tags associated with the test run.
- **JIRA bug reporting** icon - appears if Smart Reporting is integrated with JIRA. Supports entering bug reports directly as a JIRA issue.
- **Report Details** button - displays detailed information on the test run data, and device(s) data.
- **Open Support Case** - Connects directly to Perfecto Support to allow you to open a new incident.
- **Download** button - supports accessing and downloading the artifacts (video, log files) associated with the test run.

Scheduled retries

Perfecto can be configured to collapse scheduled retries into a single test report. This feature is turned off by default. To turn it on in your cloud instance, contact Perfecto Support.

For a test to be considered a retry, it must share the same parameters and CI job name and number or be part of the same execution. Perfecto does not list a test that is considered a retry in the table and does not take it into account when calculating statistics. Only the last test in a retry series makes it into the statistics.

If the current test is part of a scheduled retry, this is indicated by two arrows surrounding the double-ringed current-run icon (). In this case, the first header line includes the retry navigation bar (

). You can use this bar to scroll through the individual retries.

The following video illustrates how the scheduled retry feature works.

Your browser does not support the HTML5 video element

Failure reasons

When the test script or Smart Reporting heuristics have identified a failure reason, it appears in the second line next to the test status (only if the status is **Failed**). The failure reason is one of those configured for the CQ Lab by the administrator.

If no failure reason was identified for a failed test, the **Add failure reason** button is displayed instead.

To add a failure reason:

1. Click **Add failure reason**.
2. From the list of preconfigured failure reasons that appears, select a failure reason by select **Add custom failure** reason to enter a new failure reason.

To update a failure reason:

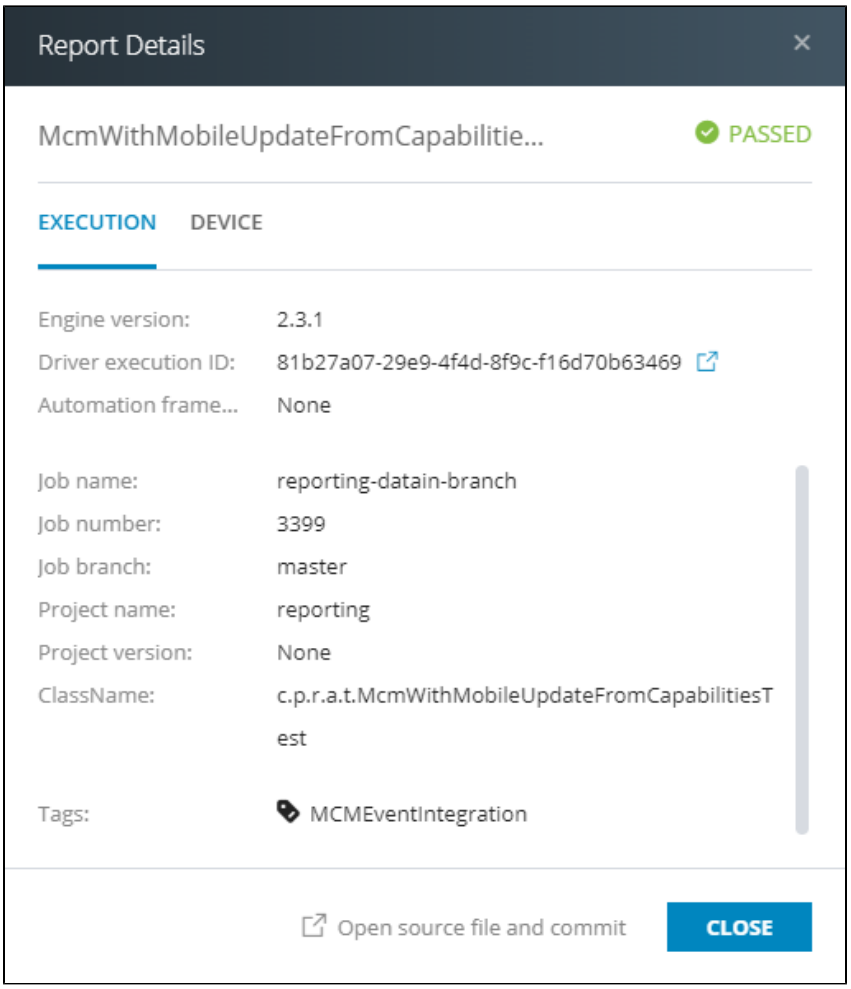
1. If you don't think the displayed failure reason is the cause of the issue, click the current failure reason.
2. From the list of preconfigured failure reasons that appears, select a different reason or select **Clear failure reason**.

> If the assigned reason is **Blocked**, you cannot update the reason.

View report details

Use the **Report Details** button  in the upper right corner of the STR view to display the **Report Details** form. This form shows information related to the selected test, as follows:

- **EXECUTION tab:** Displays data associated with the test run, including:
  - Basic execution data
  - Job information
  - Project information
  - Custom field names and values
  - Any tags associated with the run
- **DEVICE tab:** Displays information about the device used for the test. (For multiple device tests, see Tests on multiple devices below.) Information includes:
  - Device name and manufacturer
  - Device ID
  - OS version and firmware
  - Resolution
  - Location



Access source code

Sometimes, a test run does not complete as expected and results in a status of **Failed**. In such a case, it is often easier to understand what went wrong or what needs to be fixed in the test script if you can view the source code. This functionality is dependent upon the tester supplying the information, as described in Access source code. Perfecto displays source code in a new browser tab.

Links to source code are configurable via custom fields set by the test run.

- The **Open commit** link displays if the test run sets the **pefecto.vcs.commit** custom field.
- The **Open source file** link displays if the test run sets the **pefecto.vcs.filePath** custom field.
- The **Open source file and commit** link displays (but appears inactive) if the test run does not set any of these custom fields. Moving the pointer over this link opens a tooltip encouraging you to set the custom fields for future test runs.

If the test run sets both custom fields, both fields display (**Open source file** and **Open commit field**).

To open the source code from the STR:

- If the STR displays an error message, do the following:
  1. Open the full error message.
  2. Below the message, click **Open source file**.
- For all STRs:
  1. Click the **Report Details** button .
  2. At the bottom of the Report Details form, click **Open source file**.

Download artifacts

You can download a PDF version of the test report to your local workstation through the download menu, accessible from the download button at the top right.

Test reports include the basic information on the test execution displayed in the STR. In addition, the test may attach other log files as artifacts, such as of the device activity, network activity, or other vitals information.

Application crash report

There are times when a test fails because the application under test crashes. This is actually a case where the test succeeded in uncovering an application fault. When this occurs, the mobile operating system generates a crash log that includes information that the application developer can then use to identify the cause of the crash.

The Perfecto system identifies these situations, retrieves the crash log from the device, and notifies the Smart Reporting analytics of the status. Smart Reporting identifies the failure *reason* for this type of test as **Application crashed**. It add the crash log as an artifact to the test report.

This is supported on all iOS versions, Samsung devices running Android 7.0 and later, and other Android devices running Android 6.0 and later.

To retrieve the crash report for the test:

1. In the upper right, click the download button  for the test report.
2. From the menu, select **App-Crash-Report** to download the log file.

Tests on multiple devices

When a Perfecto Native Automation test script allocates multiple devices to run the test, the reporting system gathers artifacts (screenshots, video) from all of the devices involved. At the completion of the test execution, the report for the test run generates a single report that includes the artifacts from all devices.

Note: Displaying a single report for multiple devices is available **only** for Perfecto Native Automation.

Multiple devices in the Report Library view

The Report Library view lists test runs that activated multiple devices with the following indications that multiple devices were involved:

- Platform type (**Form factor**) column: Displays the number of devices used after the form factor icon (for example: )
- **Device** column: Lists all devices used
- **OS** column: Lists all OS versions used, corresponding to the devices listed
- **Resolution** column: Lists the device resolution for each device used

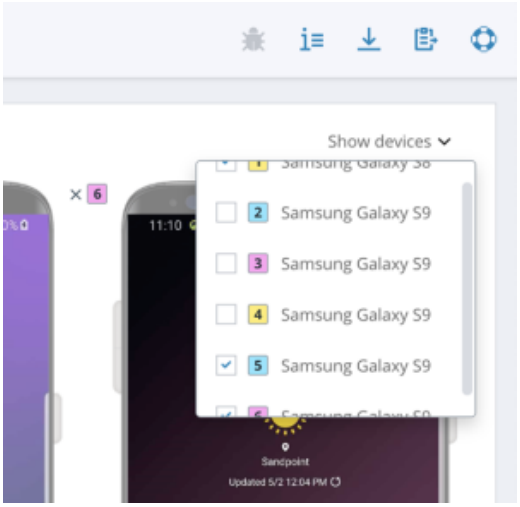
Multiple devices in the STR view

The STR of a test that activated multiple devices includes the following indications that multiple devices where involved:

- The device button on the test status line indicates the number of devices involved in the test run. Hovering over the button will open a tooltip that indicates the names and OS version of the devices involved.
- The **Report Details** form includes a dedicated tab for each device involved in the test run.
- The video shows *all* devices involved in the test run, but it only displays 3 devices at a time. You can control which devices to display (or hide) using the **Show devices** menu on the right, as shown in the following image. For devices you want to display,

select the respective check box. For devices you want to hide, clear the check box.

In addition, you can use the close button (x) at the top right of a device to stop displaying the individual device.



Screenshots are available for all devices involved.