

# Debugging Printing Issues

nainar@ | 20/2/2017

## Motivation

There tends to be a lot of back and forth between the Print and the Rendering team - Style, Layout and Paint - about where an issue lies. It is difficult to understand whether an issue lies in the Print Preview logic or the Rendering logic. This document uses some of the tools available to us to help solve this issue. For instructions see the Details section below.

## Relevant information

Andrew Gozzard (gozzard@), a Blink Style team intern worked on creating a PrintBrowser mode (design document [here](#)), wherein all pages are displayed as though they were going to be printed. The implementation can be seen [here](#). This system is useful because:

1. It allows us to write a testharness test for print issues that you can open in your browser. Earlier you were restricted to testing this in testharness only using Internals.
2. It allows us to visually see the active print media queries with pagination (which DevTools > Emulate CSS Media doesn't allow) without the overhead of Print Preview. Note: Another reason why DevTools > Emulate CSS Media is not useful is that has a different entry into the Printing logic, which doesn't help debug issues.

Side note: Andrew Gozzard also collaborated with me on writing an [explainer](#) of how the Print Preview and Rendering logic work together.

## Details

Determine the problem exists:

1. Download the minified test case (use the example in the [Appendix](#) of this document )
2. Load the page in Chrome
3. Trigger Print Preview.
4. Notice the print layout in accordance with the reproduction instructions provided in the bug. If you can't reproduce the bug at this stage, you can stop following the instructions here. (For the bug see [here](#). For the output see the [Appendix](#).)

Determine if the issue is in Print Preview or Rendering

1. Run Chrome Canary with the `--enable-print-browser` flag and the file to open

2. Notice the print layout now. (For the output see images in [Appendix](#).)
3. Conclusion:
  - a. If the layouts are both incorrect then the issue lies in the rendering logic. (Both outputs are incorrect as the expected output should not have the page numbers overlayed as they currently are.)
  - b. If the layouts differ in output, there is reason to believe that the issue lies in the Print Preview logic
4. Label the bug accordingly

## Future work

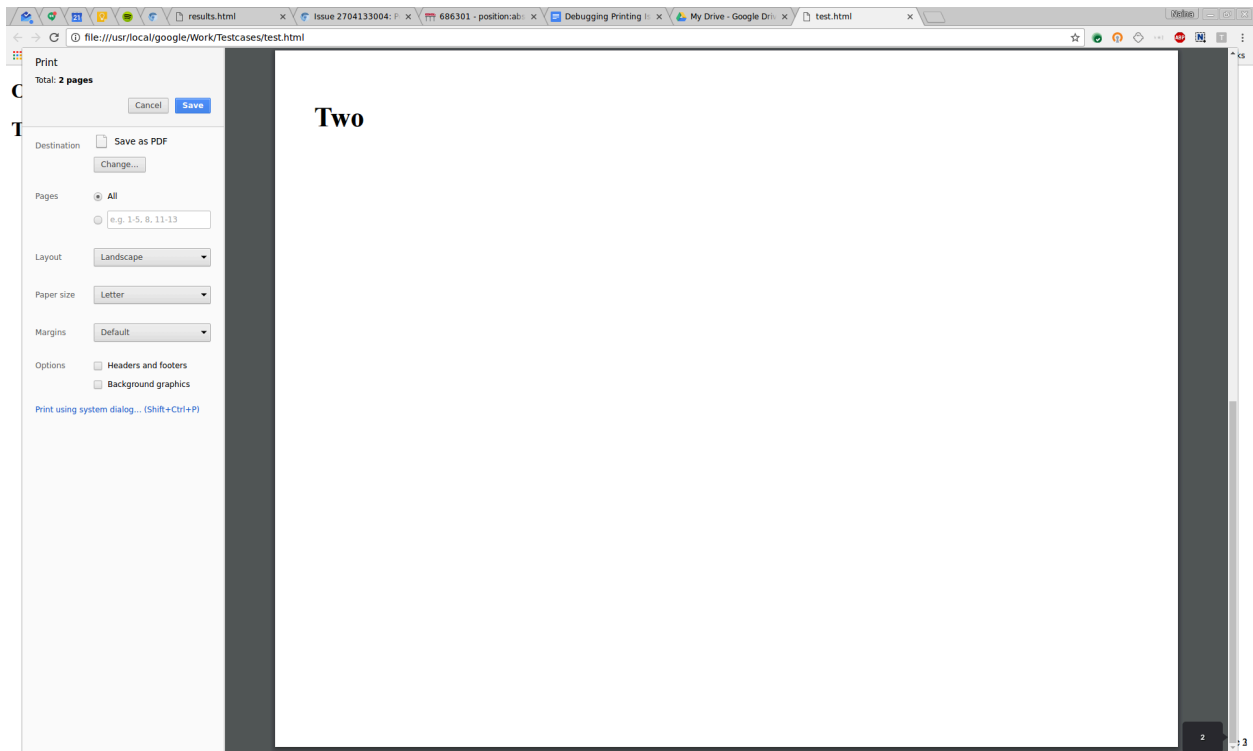
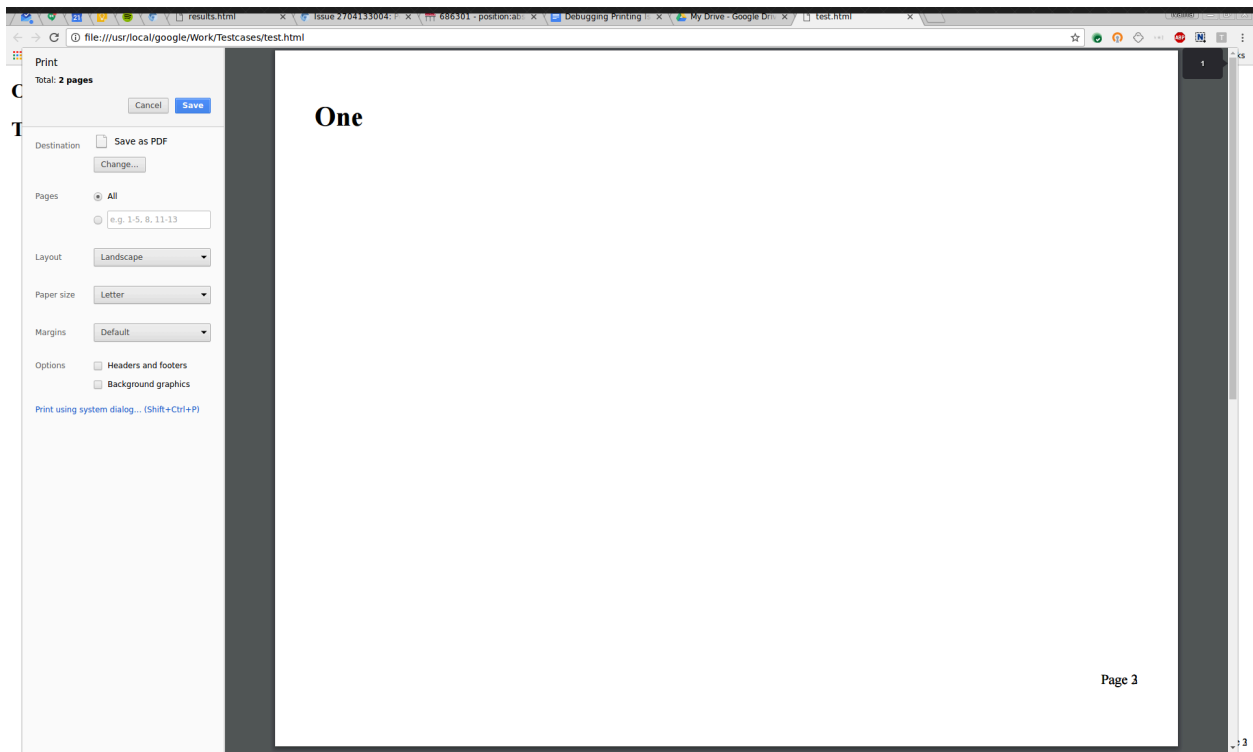
1. Being able to write more complicated tests with testharness.js
  - a. Allow Internals functionality outside Internals (page count, page size, etc)
  - b. Being able to define the page size.
2. Automate the system above to allow us to differentiate between Print Preview and Rendering issues. (Research into systems that allow you to read information from PDFs - is it possible?)

# Appendix

## Sample page

```
<!DOCTYPE html>
<html>
  <head>
    <style>
      body {
        counter-reset: page 1;
      }
      h1 {
        page-break-before: always;
      }
      .footer:after {
        bottom: 2em;
        position: absolute;
        right: 10px;
        content: 'Page ' counter(page);
        counter-increment: page;
      }
    </style>
  </head>
  <body>
    <h1>One</h1>
    <div class='footer'></div>
    <h1>Two</h1>
    <div class='footer'></div>
  </body>
</html>
```

# Output using Print Preview



# Output using --enable-print-browser



**One**

Page 1



Page 1

**Two**

