

Re: [opensuse] RFR: Printing From Browsers

Source: <http://linux.derkeiler.com/Mailing-Lists/SuSE/2008-08/msg01545.html>

- *From:* Doug McGarrett <dmcgarrett@xxxxxxxxxxxxx>
 - *Date:* Thu, 21 Aug 2008 20:52:29 -0400
-

On Wednesday 20 August 2008 23:34, Brian K. White wrote:

----- Original Message -----

From: "Randall R Schulz" <rschulz@xxxxxxxxx>

To: <opensuse@xxxxxxxxxxxxx>

Sent: Wednesday, August 20, 2008 3:53 PM

Subject: [opensuse] RFR: Printing From Browsers

Hi,

I am almost always frustrated by the results of printing HTML pages from Firefox (I'm using version 2.0.x). The most common problem I encounter is that text lines mean nothing to the pagination process and it is common for a row of text to be sliced horizontally and split across a page boundary.

Does anyone know of a browser that does well with printing?

I never heard of a browser that prints very well.

The obvious reason is html just isn't designed for that.

It's designed intentionally to flow to fill the shape of whatever container it's in. (render as sensibly as possible in any size/shape display device.) whereas documents intended to be printed are written and formatted for a specific type of "display" (hard copy) from the outset.

But that very statement about flowing to fit any window makes me think that it should be no problem for a browser or plug-in to simply re-render the page in an imaginary window that has the same properties as a printed sheet, (A4 or Letter dimensions at 300 dpi, or maybe 100dpi for rendering the page since that's what web designers assume screens are, and then scale the result to 300 or just ask the printer driver to do so)

In my own app I have to generate html specially for printing. There are a few css attributes that can provide

Re: [opensuse] RFR: Printing From Browsers

the crudest of ability to make an html document that paginates sort of ok. I have to have pre-formatted text (captured print-to-file from an application) and then in html I specify the font type and size, and wrap the text (or every x lines of it) in pre tags, and apply css page-break-before to the pre tag. Similarly, for scanned documents, I apply the same css page-break-before attrib to an img tag and display the img with 100% width and the height unspecified, which, as long as as the image was a letter or a4 geometry, ends up printing sensibly even though you can only see part of the image actually in the browser window.

Since I have to both a) have content that I somehow know is al