Webconverger wiki/
printing

[CTRL]+[p]

Webconverger utilises CUPS 1.5 which auto-detects CUPS <1.6 servers using "BrowsePoll" for CUPS Browsing for sharing and discovery. This "auto setup" feature is removed in CUPS 1.6.

Holding down the [Control] and [p] keys on your keyboard gives you the printing dialog. Or you can use the `showprintbutton` API to show a print icon to your users.

Mozilla Firefox printing guidelines can help you improve prints.

If you do **not see your printer in that dialog** or you just see "Print to File", *your* print server is **not correctly configured**.

So unfortunately the user experience of just using detected "auto setup" printers is no longer supported in CUPS. You now need to "Add printer..." up front, and hence we have introduced the `printer=` API for you to do this.

printer= API

the printer= API has syntax like so:

```
printer=printer name,device URI,driver URI
```

For example a Ricoh Aficio SP 4310N with PPD on IP 192.168.1.5 might be setup like so:

```
printer=lab1,http://192.168.1.5/printer,http://ppd.webconverger.com/ricoh.ppd
```

Instead of specifying a remote PPD, you could use the internal (generic) drivers `lpinfo -m` list.

For example a network printer 192.168.1.10 named "helpdesk" with dns name printer is very likely to have "Laser printer" PCL support:

```
printer=helpdesk,socket://printer,drv:///sample.drv/generpcl.ppd
```

However if your printer supports PostScript, you might find using the "Generic PostScript Printer" driver `drv:///sample.drv/generic.ppd` result in better quality prints.

You can have multiple `printer=` lines and hence give your users several printers to print to.

Two typical printing configurations

Using a network printer

Buy a printer with a **network port**. These printers are generally more expensive and *generally* have IPP support built in. Some do not have an IPP print server or are simply buggy. So make sure you purchase it with a good exchange / refund policy.

http://www.cups.org/documentation.php/network.html

Discovering the "device URI" can be alarmingly difficult. Usually socket://$PRINTER_IP_ADDRESS will probably work.

Using a CUPS server with (attached) printer shared upon it

Buy a printer and attach it to a MacOSX / Ubuntu / Windows PC. Proceed to install drivers and most importantly the CUPS

print server software. Then configure the CUPS server to be shared upon the network.

Add your printer to the client using the printer= API, using the an appropriate device URI.

prefs= API to remove default header and footer

The prefs= API allows you to override Firefox preferences.

For example `prefs=http://ppd.webconverger.com/2013/ricoh.js` removes header/footer for the printer named 'Ricoh'.

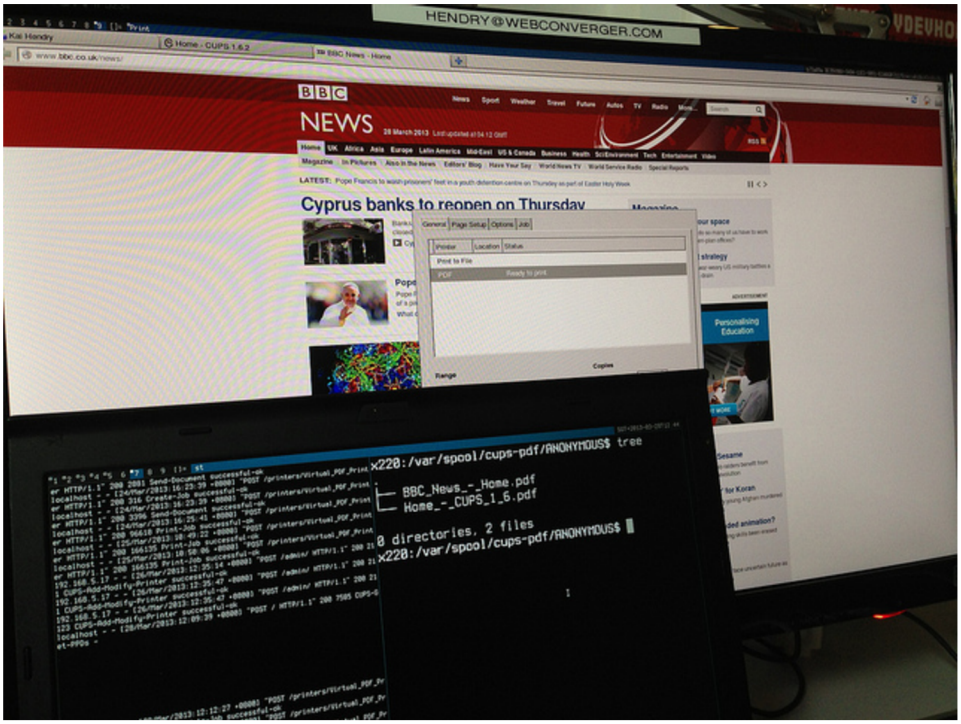
A pref such as:

```
pref("pdfjs.disabled", true);
```

Disables the default Mozilla PDF viewer, which should default to epdfview viewer. PDFJS has issues like:

- https://bugzilla.mozilla.org/show_bug.cgi?id=811002
- https://bugzilla.mozilla.org/show_bug.cgi?id=844090 also see #146

Printing to a PDF file to a remote machine on your network



The cups-pdf package allows you to setup a "Virtual Print" server on your network that write out PDFs on your server's spool directory.

This is useful for archiving, debugging and allowing you to manually check what does and does not get printed.

Assuming you have a permissively configured CUPS server (not Webconverger, an Ubuntu install for example) with cups-pdf installed, you need to figure the IP address or name (e.g. 192.168.0.1) of your print server, that can be accessed by your clients. To find the device URI of your CUPS printer, it's the "Queue Name" URL in the Administration Tab → Manage Printers, see the highlighted device URI screenshot.

Once you know the printer's device URI, you then setup your Webconverger clients like so:

```
printer=PDF,http://192.168.0.1:631/printers/Virtual_PDF_Printer,drv:///sample.drv/generic.ppd
```

On the print server, you should be able to see "Send-Document successful-ok" whilst viewing the log like so `tail -f /var/log/cups/access_log`.

Advanced debugging

https://fedoraproject.org/wiki/How_to_debug_printing_problems

Assuming you are running a Virtualbox instance, sharing a bridged adaptor. On the host 192.168.1.3 with the Virtualbox on 192.168.1.136:

```
sudo tcpdump -w /tmp/foo.pcap -s 0 -n -i wlan0 dst 192.168.1.136 and not port 53
wireshark -r /tmp/foo.pcap
```

Last edited 10 days and 7 hours ago