

Insert catchy name involving CUPS here

Matthew Geddes
Tellurian Pty Ltd

AUC X World II, July 2004

Abstract

This paper aims to cover give the reader a general overview of the CUPS printing system found on many Unix and Unix-like platforms today. This paper also focuses a little on CUPS under Apple's OS X operating system.

Contents

1	Introduction	4
1.1	What is CUPS?	4
1.2	CUPS features	4
1.2.1	IPP	4
1.2.2	Compatibility with other print systems	4
1.2.3	Included web configuration tool	4
1.2.4	Modular architecture	4
2	Printing job flow	5
2.1	User applications	5
2.2	Print scheduler	5
2.3	Print filter or Print driver	6
2.4	Print device	6
3	Printer Management	7
3.1	Log files	7
3.1.1	page_log	7
3.1.2	access_log	8
3.1.3	error_log	8
3.2	Configuration files	9
3.2.1	/etc/printcap	9
3.2.2	/etc/cups/cupsd.conf	9
3.2.3	/etc/cups/printers.conf and /etc/cups/classes.conf	9
3.2.4	/etc/cups/ppd/*	10
3.2.5	/etc/cups/client.conf	10
3.3	Command-line tools	10
3.3.1	lpstat	10
3.3.2	lpadmin	11
3.3.3	lpoptions	11
3.3.4	lpinfo	11
3.3.5	lppasswd	11
3.3.6	lpr, lprm and lpq	12
3.3.7	enable and disable	12
3.4	Using the web interface	12
3.4.1	Administration	12
3.4.2	Help	12
3.4.3	Jobs	13
3.4.4	Classes and Printers	13
3.5	Access control and resource management	13
3.5.1	Access Control	13
3.5.2	Authentication	14
3.5.3	Quotas, limits and accounting	15

1 Introduction

1.1 What is CUPS?

CUPS is an open source print system for Unix. The name CUPS actually stands for Common Unix Print System. CUPS is shipped with Apple's OS X.

1.2 CUPS features

1.2.1 IPP

CUPS supports the Internet Printing Protocol, or IPP. IPP is also supported under platforms such as Novell's Netware and some recent Microsoft Windows operating systems. Because it is widely supported and well-documented, IPP alone comes close to solving one of the biggest administration nightmares on most networks - cross-platform printing. Some higher end network printers now support IPP too.

In addition to the sending and receiving of print jobs, the IPP protocol also supports features such as *browsing*. Browsing allows the automatic configuration of printers on a network. An IPP host may broadcast it's information about it's printer list to a particular network. The other IPP hosts on the network may then add this printer information to their printer list. A common example of this is plugging in an IPP capable print device and seeing it appear in the list of network printers on all of the desktop PCs in an organisation. Browsing may also be configured such that it doesn't adversely affect large networks.

1.2.2 Compatibility with other print systems

- BSD and SysV print systems over network and command-line tools
- Windows print system with the help of Samba
- Socket printing for printing to HP JetDirect and similar devices

1.2.3 Included web configuration tool

Because the IPP protocol is very similar to the HTTP protocol, it was quite easy for the CUPS team to write a web-based configuration tool into the cupsd service. This allows you to configure and maintain any CUPS host's print queues from any machine that has a web browser.

1.2.4 Modular architecture

CUPS is designed to be very modular. To add new functionality, such as better accounting or support for a new network print protocol, it is really just a case of somebody writing a small component that serves the intended purpose and plugging it into the CUPS print system.

Apple have added their own modules to the CUPS print system. Some of Apple's extra modules include support for multicast DNS (part of the rendezvous suite of protocols) and the ability to print to bluetooth and appletalk printers.

2 Printing job flow

2.1 User applications

On OS X, we can run a few different types of applications. There are the native OS X applications that we've all become familiar with, there's the Classic applications that run on top of OS X and the Unix applications that we're also able to run. These applications can print in any number of ways.

1. In the case of OS X applications (and Classic apps in an indirect fashion), print jobs are started through the Cocoa print API. The Cocoa print API then passes the print job through to the CUPS API.
2. Legacy Unix applications print by executing one of the compatibility command-line tools (such as the `lpr` command) provided by CUPS, which in turn calls the CUPS print API
3. Modern Unix applications call the CUPS print API directly.

As you can see (and Figure 1 may help confirm this) at some point, every print job passes through the CUPS print API. From there, the CUPS print API passes the print job to the print scheduler.

The CUPS print API can be controlled by the `/etc/cups/client.conf` configuration file.

2.2 Print scheduler

Central to the print system, the print scheduler (in the case of CUPS, a service called `cupsd`) is responsible for making sure that print jobs that come in, get sent to the correct place in the correct order.

A print job is stored temporarily in a *print queue* by the print scheduler so that it may be sent to the correct print device when that device is not busy. These print queues are stored on the hard disk (usually under the `/var/spool/cups` directory). This is done to save on memory as well as allowing printing to resume across reboots.

Figure 1 shows whereabouts the print scheduler fits into the CUPS print system on OS X.

The print scheduler is controlled through a number of CUPS configuration files. The three major files you'll come across are:

- `/etc/cups/cupsd.conf`, which contains global print configuration directives
- `/etc/cups/classes.conf`, which is used to group printers together in a class or cluster
- `/etc/cups/printers.conf`, which contains printer-specific configuration items

2.3 Print filter or Print driver

A print filter (also known to some as a *print driver*) is used to turn a print job from something understood by the print system (often postscript, but PDF in the case of OS X) into something that will be understood by the print device itself. In the case of a postscript capable printer, the print job is passed through unchanged, but in the case of your typical inkjet printer, some conversion is necessary.

Under a CUPS print system, it is possible to tweak and tune the way this happens for individual print queues by modifying that print queue's PPD file. These PPD files are stored under the `/etc/cups/ppd/` directory. The format is a standard one that has been well documented by Adobe.

There are many print filter systems available. Probably the most common (and certainly the main one shipped with OS X) is called `gimpprint`. `Gimpprint` was written by the GIMP team (<http://www.gimp.org/>) to aid printing within their image manipulation application. Rather than tie it to that particular application, they made it modular enough that the CUPS (and other) teams were able to use it as a drop-in replacement for their existing print filter systems. `Gimpprint` has support for a very large number of printers.

In a network situation, the question regularly arises: "*Where should I configure my print filters/driver? On the client side or on my print server?*". Unfortunately, there's no single right answer. It does depend heavily on your particular situation. Having said that, many organisations configure all of their client machines (Windows computers especially) to print to queues on a print server using the standard, generic postscript driver (there's an excellent one available without cost from Adobe). The print server is then configured with the relevant print filters for the various print devices around your network. This is great because it provides centralised printer and driver management and requires much less maintenance on the individual desktop machines, but does add significant load on the print server.

2.4 Print device

A print device is the physical device that you plug into the power and keep healthy by feeding it a staple diet of toner and paper. Some eat more than others. Most are house-trained.

We are starting to see more and more printers with ethernet ports that are capable of receiving print jobs over a network. Some of these even support printing using the Internet Printing Protocol mentioned earlier. Many organisations configure their desktop computers to print directly to these devices, but often a more efficient way to manage this is to have all clients print to a queue on the server and then have the server feed the print jobs to the printer one at a time.

CUPS is able to print to any device using any media. All it needs is the relevant backend to send the print job (post-print filter) to the print device. Some of the backends included with CUPS under OS X (10.3.4, to be precise) are:

- Bluetooth and USB allow printing directly to a printer using one of these two bus types
- IPP and HTTP allow network printing using either the IPP or the HTTP protocols

- PAP, LPD and SMB allow printing over a network to an Appletalk, Unix or Windows print server
- The socket backend allows printing to network-capable printers similar to the HP Jet-Direct print servers
- The fax backend allows the sending of faxes
- Multicast DNS is part of the group of standards Apple markets as *rendezvous*. This backend directly relates to the 'Rendezvous' listing in the 'printer type' drop-down list in Apple's 'Printer Setup Utility'.

Parallel and Serial printers are also supported by the standard CUPS distribution from <http://www.cups.org/>.

3 Printer Management

3.1 Log files

Like other Unix services, CUPS maintains a number of different text-based log files, which can be easily read by humans (or Systems Administrators :-)) or be easily parsed by standard Unix text-processing tools (such as perl, sed/awk and cut). This section briefly introduces us to each of the CUPS log files.

These log files typically live under the `/var/log/cups/` directory.

3.1.1 page_log

The `page_log` file contains an entry for each job successfully sent to the printer. It's used for billing and accounting. By the time an entry appears in `page_log`, it's already been printed, so it's not all that useful for print quotas.

Here's an excerpt of a `page_log` file:

```
prn1 mgeddes 238 [30/Jun/2003:16:08:07 +0930] 23 1
prn1 mgeddes 238 [30/Jun/2003:16:08:07 +0930] 24 1
prn1 mgeddes 238 [30/Jun/2003:16:08:07 +0930] 25 1
```

The fields in the `page_log` are delimited by spaces. The fields are as follows:

1. The Print queue
2. The username of the user printing the job
3. A print job number (corresponds to the job number in the web-based configuration utility)
4. A date/time stamp
5. The page number
6. The number of pages

3.1.2 access_log

The access_log shows all attempts to access our CUPS service. A sample of an access_log follows:

```
localhost - - [30/Jun/2003:16:01:47 +0930] "POST /admin/ HTTP/1.1" 401 0
localhost - root [30/Jun/2003:16:01:47 +0930] "POST /admin/ HTTP/1.1" 200 8087
localhost - - [30/Jun/2003:17:13:07 +0930] "POST / HTTP/1.1" 200 140
```

The (space-delimited) fields in the access_log are as follows:

1. The hostname of the client connecting to our service. If the HostNameLookups parameter is disabled in cupsd.conf, this will contain the IP address.
2. The group field, which is currently unused.
3. The username of the authenticated user making the request.
4. The data stamp (including the offset from UTC)
5. The HTTP method
6. The resource field contains the URL path of the resource requested
7. The protocol version
8. The HTTP status code of the request (200 for success)
9. The size in bytes of the request

From the example above, we can see that all of the requests were from the machine that CUPS is running on (my workstation in Adelaide). The first line indicates (we can tell by the 401 error code and the size of 0) that authorisation failed. The browser obviously retried and gave a username and password (the username was 'root') and the request succeeded.

3.1.3 error_log

As the name suggests, error_log contains any information or errors about the way CUPS is running or being used. In the next excerpt, I've made a few formatting changes, but for the most part it's intact:

```
W [23/Jun/2003:16:41:07 +0930] Possible DoS attack - more \
    than 10 clients connecting from 192.168.1.1!
I [23/Jun/2003:... +0930] Listening to 0:631
I [23/Jun/2003:... +0930] Sending browsing info to cb1445ff:631
I [23/Jun/2003:... +0930] Configured for up to 100 clients.
I [23/Jun/2003:... +0930] Allowing up to 10 client connections per host.
I [23/Jun/2003:... +0930] LoadPPDs: Read "/etc/cups/ppds.dat", 3259 PPDs..
I [23/Jun/2003:... +0930] LoadPPDs: No new or changed PPDs...
```

The fields in the above log file are as follows:

1. Error level (W for warning, I for information, E for error and D for debug)
2. A date/time stamp (which I've butchered in the above example)
3. The message itself

In this excerpt, the first line indicates that CUPS has had an awful lot of dubious looking requests from a particular host (192.168.1.1). CUPS will eventually (hopefully before it's too late) take some kind of action. This usually consists of refusing any more connections. In this case I know that it wasn't a real DoS attack, but was some bad behaviour by the client.

3.2 Configuration files

CUPS has a number of configuration files that can each be tweaked by hand to allow fine-grained manual control over the way your CUPS printing system operates.

3.2.1 `/etc/printcap`

The `/etc/printcap` file is provided for backwards compatibility with older Unix print systems that current applications may still rely on. It is maintained by CUPS and rarely needs to be touched by mere mortals.

3.2.2 `/etc/cups/cupsd.conf`

`cupsd.conf` is where most of the global configuration happens. Some of the configurable parameters found in `cupsd.conf` include:

- Access control and security
- IPP Browsing
- Performance
- Network parameters (including SSL support)

3.2.3 `/etc/cups/printers.conf` and `/etc/cups/classes.conf`

Printer specific configuration parameters are stored here. Some of these parameters include:

- Page and Byte Limits
- Quota period
- User access control
- Printer location and description

- Printer status

It's possible to tweak some of these options by modifying the file itself, but there are command-line tools (which we'll discuss shortly) that make this task a little easier.

When manipulating printer classes, the `/etc/cups/classes.conf` file is used. It's syntax and supported parameters are very similar to `printers.conf`.

3.2.4 `/etc/cups/ppd/*`

PPD files are, in essence, a per-printer configuration file. Some of the parameters that are typically set in a PPD file are:

- The default paper size
- The printer resolution
- Whether to print color or greyscale by default
- Print filters (to convert from postscript or PDF to something understood by the printer)

PPD files are typically left alone by most systems administrators. If you can't find a PPD for the printer you're using, have a look at <http://www.linuxprinting.org/>

3.2.5 `/etc/cups/client.conf`

This configuration file is used to control the default behaviour of the client tools for CUPS. Parameters that may be set include:

- Whether to use SSL encryption or not
- The default server to connect to

3.3 Command-line tools

Most of the command-line tools shipped with CUPS are provided for backwards compatibility with older Unix print systems. As a result, the command-line options are pretty much the same, except where CUPS implements features not found in other print systems.

3.3.1 `lpstat`

`lpstat(8)` can be used to view the status of each of the currently installed printers and printer classes. For example:

```
diving:~ matthew$ lpstat -a
lp accepting requests since Jan 01 00:00
```

3.3.2 lpadmin

The `lpadmin(8)` utility is a command-line tool for manipulating and configuring printers and printer classes under CUPS. `lpadmin(8)` (like many of the CUPS utilities) has a `-h` option, which allows you to specify the host to perform the operation on. To set the default print queue on a host to `lp`, you can use a command similar to the following:

```
diving:~ matthew$ lpadmin -d lp
Password for mgeddes on localhost?
```

3.3.3 lpoptions

The `lpoptions(1)` tool allows the administrator to both get and set various printer-related options. Figure ?? contains an example of the `lpoptions` command being used to set a quota on a print queue.

3.3.4 lpinfo

`lpinfo(8)` can be used to show all available printer backends and drivers on the system. The following example demonstrates the `-v` option, which lists all available print backends. The `-m` option shows all available print filters.

```
diving:~ matthew$ lpinfo -v
network pap
network socket
direct bluetooth
serial fax://dev/cu.Bluetooth-Modem
serial fax
network http
network ipp
network lpd
serial fax://dev/cu.modem
direct usb
network smb
network mdns
```

3.3.5 lppasswd

The `lppasswd(8)` tool can be used to add, modify or delete user passwords stored in the `/etc/cups/passwd.md5` file. The `passwd.md5` file is only used when Digest authentication is being used.

3.3.6 lpr, lprm and lpq

lpr(1), lprm(1) and lpq(1) are command line tools provided for backwards compatibility that allow a user to print a file or files (lpr), remove a particular job from the print queue (lprm) or view the contents of the print queue (lpq).

Figure ?? demonstrates the use of these three commands.

```
diving:~ matthew$ lpr ~/CUPS2.ps
diving:~ matthew$ lpq -Plp
lp is ready and printing
Rank   Owner   Job      File(s)                Total Size
active matthew 4        CUPS2.ps                717824 bytes
diving:~ matthew$ lprm 4
diving:~ matthew$ lpq -Plp
```

3.3.7 enable and disable

enable and disable can be used to turn on or turn off the print queue's will to print. enable is the same as specifying the -E option as one of the last options to many of the above command-line tools.

3.4 Using the web interface

The web interface for CUPS provides browser-based access to many of the features that are provided by the various command-line tools we've just looked at. Whilst graphical interfaces are often less flexible than a suite of command-line tools, many users and even administrators prefer them.

Figure 2 is a screenshot of the Printers section of the CUPS web interface. As you can see from this screenshot, there are a number of other configuration sections.

3.4.1 Administration

The Administration section, like anything that involves making changes, requires authentication. It allows the user to add, modify and delete print queues and print classes.

3.4.2 Help

The Help section contains all of the shipped documentation for CUPS. If for some reason your distribution is missing any of this documentation, it's always available at <http://www.cups.org/>. Of particular interest to most Systems Administrators are the following documents:

- The Users' Guide
- The Administrators' Guide and

- Parts of the Developers' Guide, in particular the sections on CUPS backends and filter systems

3.4.3 Jobs

The Jobs section of the interface is used to manipulate (stop, start, restart) print jobs in our print queues.

3.4.4 Classes and Printers

The Classes and Printers sections both allow us to add, modify or remove print queues. The difference between the two is that a printer class often contains multiple print queues. We can define a number of print queues under the Printers section and group them together in a Printer class. This allows a crude form of load balancing and redundancy.

3.5 Access control and resource management

Authentication and access control under CUPS is becoming more mature all the time, but there are a few areas where it's not quite as flexible as many enterprises may need.

3.5.1 Access Control

Central to CUPS is the IPP protocol, which is similar to the HTTP protocol that we use to view the web. In fact describing the location of a given printer is done using a URL when using IPP, much like the way we describe the location of a web page using a URL. Here's an example of both a simple HTTP and a simple IPP URL:

```
http://www.apple.com/macosx/
```

```
ipp://print.apple.com/printers/lp
```

The first example is a URL to some OS X-related information from Apple's web site, whereas the second is an example of an IPP URL which points to a printer queue called lp on a machine called print.apple.com.

To be able to assign print privileges to users under CUPS, we can just tell CUPS to apply a certain type of authentication to a particular path of the URL. The following fragment of the `/etc/cups/cupsd.conf` file is an example of how to deny all but a particular subnet of machines to print to the printers on this host:

```
<Location /printers>  
  Order Deny,Allow  
  Deny From All  
  Allow From 192.168.1.255/24  
</Location>
```

This particular example tells CUPS that anything under the `/printers` path is only accessible to those computers on the `192.168.1.255/24` subnet.

Because CUPS implements the IPP protocol, which is quite similar to HTTP, it was a relatively easy job to build a web-based configuration interface into CUPS. You can access this web interface by pointing a web browser at the URL:

```
http://localhost:631/
```

You'll notice as you browse around that the URL path changes to reflect the particular section of the interface you're currently browsing. For example, when you click on the 'Classes' sub-menu, the URL path changes to `/classes`. By itself, not all that exciting, but coupled with what we already know, we can probably figure out that it's possible to assign the same kinds of privileges to these sections and subsections of the configuration interface.

3.5.2 Authentication

As you were browsing around the configuration interface in the last section, you may have noticed that attempting to access the 'Administration' section caused an username/password box to appear. As well as host-based access control, as we saw in the last section, it's possible to use user-based access control. Let's take another look at that previous fragment of the `/etc/cups/cupsd.conf` file, but with one form of user-based access control enabled.

```
<Location /classes>
  Order Deny,Allow
  Deny All
  Allow 192.168.1.255/24

  AuthType Basic
</Location>
```

The 'AuthType' parameter is used to tell CUPS which type of authentication to use for the particular Location. Possible authentication methods are:

- Basic - Uses PAM for authentication. OS X also comes with a PAM module that allows PAM-based authentication requests to be passed through to the native Apple NetInfo/OpenDirectory authentication system.
- Digest - MD5 Digest authentication (not yet supported by all browsers)
- x509 certificates such as those used by SSL-enabled web sites

By default, CUPS will authorise anyone supplying correct credentials for an account that is a member of the `sys` group. This behaviour can be altered to allow only members of the `printers` group (for example) by also adding the `AuthClass Group` and the `AuthGroupName printers` parameters.

3.5.3 Quotas, limits and accounting

CUPS has an amount of accounting built into it. It can apparently be fooled by a well-crafted piece of Postscript, but this is the case for many printer accounting systems.

To set a simple quota of 3 pages per-hour on a print queue called 'lp', using the `lpoptions` command, you could use a command similar to the following:

```
lpoptions -p lp -o job-page-limit=3 -o job-quota-period=3600
```

If the CUPS printer accounting is not up to the standard you require, there are a few other accounting and billing applications available.

- PyKota - http://www.librelogiciel.com/software/PyKota/action_Presentation
- PrintBill - <http://ieee.uow.edu.au/~daniel/software/printbill/>

4 References

- <http://www.cups.org/> - The CUPS web site
- <http://developer.apple.com/printing/index.html> - The Apple printing site
- <http://www.linuxprinting.org/> - contains a large amount of information regarding printing on Linux using many print system (including CUPS). Of particular interest is the printer database.



Figure 1: CUPS print job flow

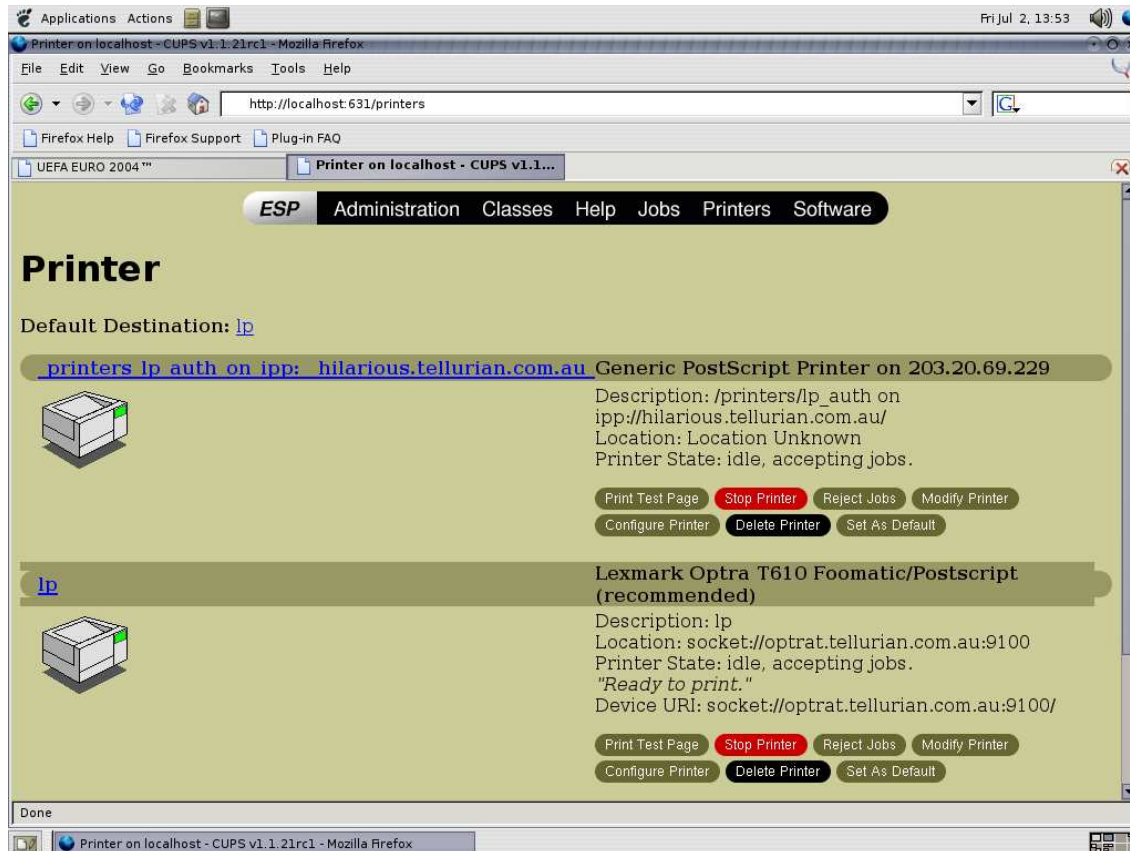


Figure 2: The printers section of the CUPS web interface