Managing
Network Printers and Print Spoolers

Patrick Powell

papowell@astart.com

Astart Technologies, 9475 Chesapeake Dr., Suite D

San Diego, CA 92123

619-874-6543 http://www.astart.com

Part 1- Printer Hardware and Firmware

- Printer Mechanisms
- Host/Printer Connections
  - Serial, Parallel, Network
  - Configuration and Problems
  - Network Printer Support
- Print Job Formats for Vintage Printers
- Print Job Formats for Modern Printers
- Page Description Languages
  - PostScript and PCL
- Job Control and Format Languages
  - PostScript, PCL, and PJL

Part 2-BSD Print Spoolers

- Print Spooler Basics
  - Print Spooler Client-Server Structure
  - BSD, SVR4, Novell, MS
  - RFC1179
- BSD Print Spoolers
- LPRng
- Printing Process
- Printcap Files
- Filters
- Printcap Examples

Part 3 - Managing Print Spooler Operations

- Management Facilities
  - lpc, printcap
- Adding New Printers
  - checkpc (lprng)
- Installation
  - replacing print spooling system
- Diagnostics
- Load Sharing and Printer Pools
- Bounce Queues
- Routing
- Host Specific Printcap Entries

Part 4 - Horrible Problems

- Permissions
- Security and Authentication
- Accounting
- SNMP

Part 5 - Multi-platform Printing and Gateways

- Network Print Spooling
- Drivers and Print Spoolers
- Printer Gateways
- Microsoft Print Spooler
- Windows 95 Wslpr

Part 5 - Cont'd

- NT LPR Support
- Novell Print Spooler
- PCNFS (Sun Microsystems)
- PCNFSD LPR Support
- Samba
- Samba LPR Support
- Desperate Measures

References

- PostScript Tutorial and Cookbook, Adobe Systems, Addison-Wesley 1985
- PostScript Language Program Design, Adobe Systems, Addison-Wesley 1985
- PostScript Language Reference Manual, 2nd Edition, Adobe Systems, Inc. Addison-Wesley, 1990
- Portable Document Format Reference Manual, Adobe Systems, Inc. Addison-Wesley, 1993
- Web Site for Adobe and more documents:

http://www.adobe.com/supportservice/devrelations/technotes.html

Software

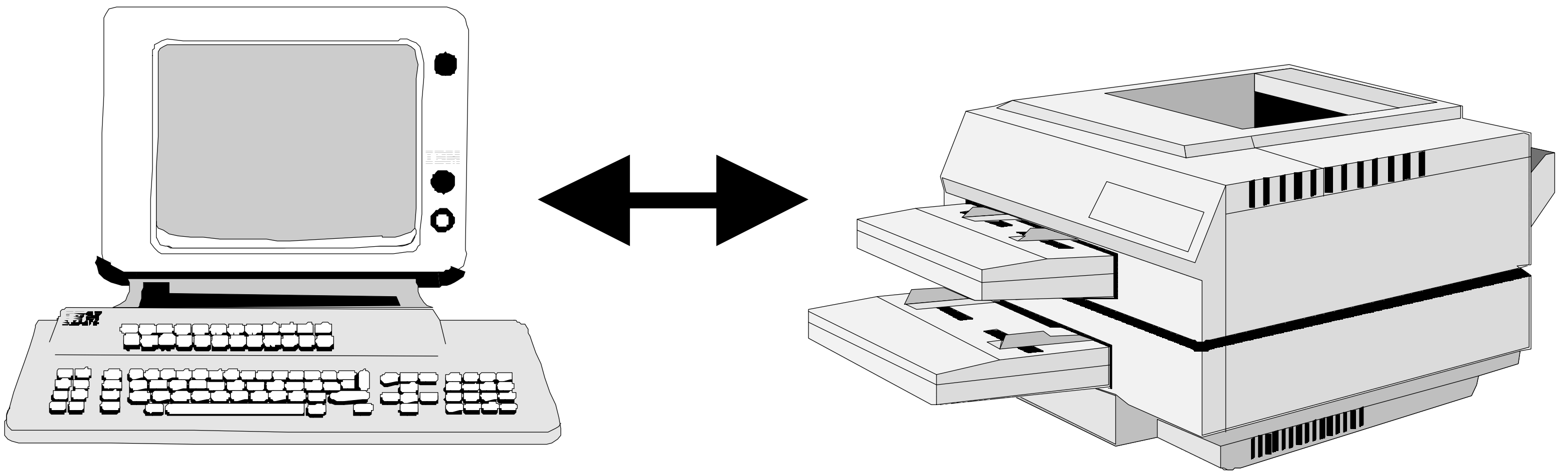
- LPRng ftp site: unless otherwise indicated, all software can be obtained from the ftp://ftp.astart.com/LPRng site.
  See the README and INDEX file
- GNU software can be obtained from many mirror sites. Try ftp://ftp.uu.net/pub/gnu and look for MIRRORS

PostScript Utilities	psutil31.tgz
More PostScript Utilities	psutilmore2.tgz
GhostScript	GNU
GhostView	GNU
Adobe Acrobat PDF viewer/tools	www.adobe.com
LPRng	LPRng distribution
FILTERS_LPRng	collection of filters, patched for LPRng

Part 1

Printer Hardware and Firmware

Basic Printer Operation



- A Printer is a peripheral device, usually attached to a host computer
- The host computer transfer print files to the printer over the communication channel

Printer Communication Channels

- Printer must be connected to source of print jobs
- Channel determines the rate at which text (characters) are delivered to printer
- Also determines the reliability
- Also determines the availability of error messages and diagnostics
- Simple and Cheap is not always best

Serial Port

- Slowest and most error prone
- Older printers had a very slow transfer speed (9600 bits/second) due to the real time requirements of handling paper
- Newer ones can go up to 115 Kbps
- Data transmissions errors VERY common
- Printer does not have a large buffer to store input so flow control is absolutely required
- Hardware (RTS/CTS) flow control almost mandatory
- Software flow control (DCON/DCOFF or CTRL-S/CTRL-Q) can be used over networks
- Watch out for buffering in network!

Advantages of Serial Port

- Error and Status messages available from printer
- Most systems have multiple serial ports, can attach multiple printers
- Can be put on terminal server (watch out for flow control, enable RTS/CTS)
- Cables can be up to 50 feet long
- Cheat and they can be up to 200 feet long

Disadvantages of Serial Port

- Very very slow
- Did I mention flow control problems? This is a major headache
- Errors in data transmissions can have interesting effects
- Note: Software Flow Control Headaches
  - Some software flow control tries to accommodate errors by restarting transmission when ANY character is received from the other end after a time-out period.
  - When you get a printer error, the printer sends CTRL-S to stop job and then later reports status, you lose rest of job. (Headache #27)

Parallel Port - Advantages

- Transfers data 8 bits at a time
- Flow control implicit because it uses a hardware handshake for data transfer.
- Very low error rate on data transfer

Parallel Port - Disadvantages

- Has limited bi-directional capability
  - Out of Paper, Error indication
- Bi-directional support limited on most interfaces
  - May not be able to indicate error condition when flow control enabled
- Surprisingly low throughput due to interrupt per byte on most PCs and workstations
  - Games played at OS level to improve this
- Real pain is getting status back using bi-directional mode
- Real cheap folks can even use print sharing boxes
  - You get what you pay for
  - Don’t call me when they lock up

Network Interface

- Usually supports various protocols
  - IPX (Novell and Microsoft)
  - NETBUI (Microsoft)
  - TCP/IP (LPD + RAW Connection)
- Printer builders are (or did not use to be) network aware
- Has changed tremendously over last couple of years
- Configuration of interface is now usually pretty simple once you understand the various options
- Front panel configuration is really quite simple
- HP Jetdirect has BOOTP/DHCP configuration

Network Disadvantages

- Anybody can print on the printer
  - Leads to the mysterious *print job from nowhere*
- In spite of being on the network, only one person can actively use the printer at a time
  - Yes, yes, I know what the documentation says and I am telling you what REALLY happens
- Multiple users can/will/have locked up the interface (this is a known problem with HP Jetdirect interfaces)
- Multiple users trying different protocols can/will/have locked up the interface
- When the printer dies there is usually no handy way to reset it without powering it down

Network Advantages

- Very high throughput, and has built in flow control
- Very low transfer error rate
  - Ethernet has CRC
  - Higher level protocols also do checking at transport level
- Can be configured using BOOTP/DHCP
- Status can be obtained by using SMTP on most systems
  - If DHCP working, can even reboot printer
- Printer sharing becomes very simple, in principle
- For shared system resource, may actually be the cheapest interface as it does not require host for support (see LPR/LPD later for details)

Hybrid Solutions

- Print Server Boxes
  - Has network interface, supports parallel port and serial ports for printers and/or modems
  - I have used several different ones with various levels of success
    - Lantronics - works, configuration tricky, good functionality
    - Rose - works, configuration simple, limited functionality
- Dumb Network Interface (LPserver UNIX/NT/W95)
  - Extremely stupid program that turns your PC into a TCP/IP raw socket connection to the printers serial (bi-directional) or parallel (unidirectional) port
  - This has its place when doing accounting, very tricky printer operations

HP Jetdirect Configuration

- Similar in principle to most printer network interfaces
- Front Panel Configuration
  - Enable basic network protocols
    - IPX/Novell Print Spooler
    - DCP/Microsoft
    - TCP/IP
      - Set IP address, netmask, syslog server
    - Enable BOOTP/DHCP configuration
- BOOTP/DHCP Configuration
  - BOOTP/DHCP supplies IP configuration information
  - Specifies a TFTP server and file that has detailed printer configuration information

Jetdirect Configuration File

 # Administration Info

```
name: picasso
location: 1st floor, south wall
contact: Phil, ext 1234
```

 # Only allow connections from network/netmask or host

```
allow: 10.0.0.0  255.0.0.0
allow: 15.1.2.3
idle-timeout: 120
syslog-facility: local2
```

 # SNMP Configuration

```
get-community-name: blue
set-community-name: yellow
trap-community-name: red
trap-dest: 15.1.2.3
trap-dest: 15.2.3.4
authentication-trap: on
```

Jetdirect Restrictions

- The “allow” configuration parameter enables you to restrict access to the printer from the specified network/subnetmask addresses
- This is essential to controlling access to your printer
- You need to use the BOOTP/DHCP to set the IP address, gateway, and syslog server
- Note: Setting up the BOOTP/DHCP server can be tricky. You will need to either have a DHCP server on each subnet, run a ‘forwarder’ process on a host on the subnet, or have your router forward requests to a server.   (Hint: if you have Cisco router, use the “ip helper” command to specify the DHCP server address.)

Direct Connections

- TCP/IP Port 9100 on the Jetdirect card is a direct connection to the print engine
- This is a bi-directional connection, and error messages will be written on it
- Other network support cards have similar facilities; if they do not, then DON'T USE THE CARD
- Usually only one connection at a time can be active; this can cause problems if trying to share the printer among several different hosts

SNMP

- Simple Network Management Protocol provides a common interface to obtaining information about the status of network devices.
- An “agent” process resides in the network device, spies on the activities of the device, and provides information when queried.
- Agents can also generate messages (traps) when a specific activity is detected
- A “manager” process queries agents for status
- Agents can also “write” information to the network device when requested by a manager. This allows configuration management to be done by a manager.

SNMP To The Rescue (Maybe)

- The SNMP standards include a Printer MIB
- In principle, you can use a simple SNMP manager to query the values of the Printer MIB. These are, to put it mildly, very basic.
- In practice, most vendors have extended the MIB to provide more detailed information about the error conditions, status, etc.....
- Unfortunately, most print spoolers do not use the SNMP facilities to query printers
- The common SNMP managers such as HP Openlook, SunNetManager, etc..... have the common printer MIBs already provided.
- If you are an administrator, please learn more about managers

Print Job Formats
and
Page Description Languages

OR

What Do You Send To The Printer?

Print Job Data Formats

- Coherent and organized approach to this subject is impossible due to historical development
- Will give a rapid, functional, biased, opinionated, subjective, etc.... view of the subject
- Remember: each manufacturer tries to distinguish their product in the market
- Remember: THERE ARE NO STANDARDS
  - Actually, this is a lie. There are standards. Nobody follows them :-)
- Remember: The printer with the largest market share becomes the standard
- Thus: *Every 3 years there is a new standard.*

Vintage Printers (Impact, Daisy, etc...)

- Fixed size characters (Fixed width/height fonts)
- Character set at the whim/market demand of the manufacturers
- Page dimensions based on paper sizes
  - 8.5 x 11 inches (letter), 11 x 15 (computer listing)
  - some larger size (legal)
  - whacko A4/A3 sizes (Europe)
- De Facto Standard
  - 132 columns and 66 lines
  - Unless it was 60 lines (margin at top? bottom)
  - Or 80 columns with 66 line???
  - And don’t forget metric sizes...

Printing Text on Antiques

- Send characters, print characters. Simple? Wrong.
  - You forgot about *INTERNATIONALIZATION*
  - $\$\Upsilon$ are simple, try typesetting French, etc....
- How about EBCDIIC? (Don't laugh)
- Tabs? Support for tabs? Why?
  - Smaller text files, good for limited file systems
  - Harder to change to other manufacturer's printer
  - De Facto: tabs are at 8 positions
- Market forces demand ***features***
  - *Italics*, true **bolds**, Condensed Fonts
  - Need to have *escape sequences* to enable these
  - THERE ARE NO STANDARDS

Advice on Antiques

- There are a lot of them out there - they were built for abuse, they run and run and run
- Get the manuals, make 3 copies, and save them for the next admin - they will probably still be in use when you retire
- Most new impact printers are extremely simple to use
  – Very few features
  – Modeled on major (antique) market dominators
- If you have to print multiple copies on forms, this is your ONLY choice, so make sure you choose wisely
- Daisy wheels make best multiple copy impressions
- Dot matrix are faster, wear out
- Keep a spare in the back room for parts

Plotters

- OK, these are not printers, but they turned out to inspire the next step in Page Description Languages
- Original plotters were Analog, hooked to instrumentation.
  - Drew nice curves on EXPENSIVE green paper
- Market developed for plotters which could be attached to ‘smart instruments’ or computers which simulated analog plotters
  - Drew backgrounds, labels, etc...
- HP (and others) developed several Plotter Control Languages
  - Had embedded commands for drawing text

Raster Output Devices

- Originally electrostatic plotters, would draw a plot a line at a time
- Needed to convert Gerber (or HP or IBM…) plotter stuff to raster format
- Insight that they could also produce hard copy of text using various fonts
- TROFF meets VARIAN to produce $$$ of revenue as graduate students burn up 100s of rolls of expensive electrostatic paper

Xerography

- Xerography works by whacking charges onto a drum
- The whacking is done by high intensity tightly focused light (laser?)
- Generated by scanning original mechanically

OR BY SHINING A LASER ONTO THE PAPER AND TURNING THE LIGHT ON AND OFF UNDER COMPUTER CONTROL

VOILA!

The Laser Printer

PostScript Origins

- Developed originally as part of 1975-1980 research into computer graphics
- Origins in computer picture generation
- Text, fonts, etc..., were wedged into the language, trying to import concepts of typography into the programming language
- Model used was to have dumb programs generate PostScript, and have a smart PostScript converter do rasterization
- Adobe produced first PostScript Language definition, and the first PostScript printers

PostScript Description

- PostScript is a Stack Based Programming Language
- You don’t want to know. You REALLY don’t want to know.
- If you want to know, you are warped. Or a Computer Science Whacko.
- PostScript files are programs.
- A typesetting program is a Program Generator that produces programs for another computer that runs to produce your output.
- Are you starting to understand why you have problems with printing PostScript?

PostScript Printer Control

- Since PostScript is a program running on the printer, why not give it the ability to control the type of paper, paper trays to be used, duplex printing, etc...?
- We do this by having the PostScript program set values of various system dictionary variables (I told you, you REALLY don't want to know).
- After the job finishes, you should really revert to a known state of these system dictionary variables
- Right? Wrong! You may want to set these to be permanent for all jobs
- So we have a password needed. Guess what the password is? Right. You guessed it! (Answer: no password)

PostScript Disadvantages

- Since it is a programming language, it comes in different versions
  - This is mandatory, and is Bernard's Law: Never trust Version 1.
- It requires memory to store the program, to hold temporary results, and to do rasterization, font conversions, etc....
  - Thus we encounter Booth's Law: You never have enough memory for a graphics program.
- There is always a bug in the compiler/interpreter
  - Even correct PostScript code does not always work correctly when the interpreter has errors

PostScript Disadvantages (Cont'd)

- Since you are running a program, the page generation can be hideously slow if you are doing fancy graphics operations such as scaling, rotation, etc... of large bit mapped graphics and fills
- When a problem is encountered, you have limited diagnostic capability.

Ghostscript Is Your Friend - Use It

Watch out for PostScript Interpreter Version/Level/Revision Problems

PostScript Advantages

- Incredibly portable across different vendors' printers for text/picture generation
- Not due to the language, but due to Adobe selling the code for the Interpreter at such a low price that everybody used it (Not true any more, so we are seeing some interesting PostScript bugs).
- Now everybody needs to be Adobe Bug Compatible
  – Remember: THERE ARE NO STANDARDS
- Previewers can be built that will give you an exact idea of what your PostScript will look like
- De Facto the standard, most portable way to generate documents for printing

PostScript Books

- PostScript Tutorial and Cookbook
  - The Blue Book
  - Very easy to read
  - Use Ghostscript and learn about PostScript programming
- PostScript Language Reference Manual
  - 2nd Edition
  - Almost incomprehensible; it IS a standard, after all
  - Appendixes are VERY helpful
  - You can get a copy from Adobe’s web site

    `www.adobe.com`

Embedded PostScript (EPS)

- Self contained PostScript files that produce a graphical object when executed
- See PostScript Reference Manual Appendix H
  - Version 3.0 is most common
  - Almost all graphical object editors import and/or export EPS
  - Some Web browsers even have extensions that display it

PostScript Document Structuring Conventions

- PostScript Reference Manual, Appendix G
  - Version 3 is most generally used
- Specifies how PostScript should be used for document generation
  - Prologue sets up overall characteristics of document, such as fonts, subroutines, etc...
  - Body consists of set of independent pages
  - Each page can be removed/duplicated/inserted in document without altering printing
  - Standard also provides guidelines for information about document

Tools For PostScript

- Assumes Document Structure Version 3
- PS Utilities Version 3.1 (psutil3.tar.gz)
  - Set of UNIX tools for massaging PostScript file
  - psnup will print N virtual pages per real page
  - psselect will select pages to be printed
  - psrev will reverse page order
- PS fixer (psmoreutils2.tar.gz)
  - merges pages of two documents, good for putting a background on a document
  - Selectively places/orients individual pages or sets of pages on a document
  - desperation tool made from psutils, cannot live without it

Tools for PostScript (Cont'd)

- Several tools available that allow you to add documentation control to PostScript files
  - For example, you can have the first page fed from a special tray, and then remainder from another tray
  - These are usually commercially available, but freeware/shareware versions have been spotted.
  - Some commercial print spoolers incorporate this functionality into their Print Spooler filters (see later)
- See the LPRng ftp/web site for pointers

Binary Communications Protocol

- PostScript language specification states that the program must be in printable ASCII characters or a limited set of control characters
- However, you can embed inline binary data into a PostScript file using very tricky methods
- Some control characters normally cause a PostScript Interpreter to end execution, send status, or may actually be ignored.
- The Binary Communications Protocol escapes these control characters. If you have character C needing escape you send: 0x01   C ^ 0x40
  – e.g. - 0x01 (^A) -> 0x01 ‘A’ or 0x01 0x41

Tagged Binary Communications Protocol

- Warning
  - If you have a file with embedded BCP escapes and you redo the BCP escapes, then you will destroy the escaping
- This leads to the Tagged BCP protocol
  - We add a ^A M sequence (this is an escaped CR character) to indicate that we are doing BCP and have put in the escapes
  - When we scan the file and detect ^A M we do not add more escapes
- This is one of the leading causes of problems printing PostScript files with binary images produced by some graphics programs

Postscript Printer Description Files (PPD)

- You can use PostScript to manage and control various printer operations
  - What variables/operations/values do you need?
- Each PostScript printer should have a PostScript Printer Description file that contains the various PostScript manageable options and how to set them
- Example: HP4MP - setting manual feed

```
*ManualFeed True: "1 dict dup /ManualFeed true
  put setpagedevice"
*ManualFeed False: "1 dict dup /ManualFeed false
  put setpagedevice"
```

- The contents include the PostScript needed to set the required operations
- Easy to extract from the PPD file and send to printers

PPD Warnings

- Each printer has a possibly different set of commands, so check the PPD file for the printers
- Manufacturers are not required to provide PPD files, so you may have to dig around for them.
- There are many extensions to the basic PPD capabilities, some of which are very specific to a particular printer.
- WARNING: I have discovered discrepancies between the PPD, manual, and actual operation. RTFM, and try it out.
- WARNING: some options interact, and will lock up printers. PPD and documents do not cover this

HP PCL

- Hewlett Packard developed a line of laser printers and wanted to sell them to the various printing markets
- They also did not like paying Adobe royalties and licensing fees
- They invented the Printer Command Language (PCL)
  - Version 1, Version 2, Version 3, Version 4...
  - Currently we have Version 5 and rumors of 6
- PCL is NOT a programming language, it is a Printer Command Language
- It specifies where on a page to draw lines, glyphs, and does it very well.

PCL References

- PCL 5 Printer Language Technical Reference
  - Surprisingly readable, but BORING, repetitive, and written in a horribly verbose manner.
- Read and memorize Section 3.8
  - Resetting the Print Environment
  - \]E and \]%-12345X are your friends, and will help you keep your sanity
  - Note that this sequence, when sent in a PCL file, will terminate job printing and may have some surprising consequences
- Read Chapter 4 - PCL Job Control Commands
  - This will explain most of the printer control functions

PCL Evolution

- Original versions of PCL did not support downloadable fonts.
- This was supposed to be a feature - you would buy font cards and plug them in.
- User feedback (flames) convinced HP that they needed to support downloadable fonts
- Then we had raster graphics support added
- Then we had page structuring support added
- It now can make the printer sing, dance, and do just about anything (as long as you know the correct PCL commands)

PCL Disadvantages

- Older PCL printers do not have downloadable font support
- You may (again) need lots of memory for fonts, images, etc.... (There's no such thing as a free lunch)
- It uses control characters (ESC) for many of the commands, making it difficult to fix/mangle/edit PCL files without special editors capable of handling long lines, control characters, etc....

PCL Advantages

- Simple simple to generate
- Slightly smaller files than PostScript
- Very much faster page generation, as there is little to do except read the input and copy BitBlits to the graphics memory
- Even has commands to do the printer mechanism commands such as bin selections, etc...., built into the language now.

Mystery PCL Commands

- Since each printer has a different set of capabilities, you need to have PCL commands to operate the printer
- There should be a “PCL Printer Description” file for each printer, right?
  - ANSWER: no
- Each printer should have a document providing a complete list of the PCL commands supported, right?
  - ANSWER: ummm… right… but only the development group has that information, the printer is not in development any more, …
- Situation is getting better, but documentation is still weak point for specialized operations

Quick Test

- If you are using PostScript and TBCP, what is the end TBCP sequence?

  ANSWER: \]%-12345X
- If you want to make sure that your PostScript job gets printed, even if the previous job did not end with the EOJ string, what should you put into your file?

  ANSWER: \]%-12345X before the PostScript
- If you are trying to decide if a file is PostScript or PJL or text, you might try looking for %!PS as the first characters, right?

  ANSWER: Yes, No, Maybe? Sigh… It all depends on how smart/stupid/weird the writers of the PostScript generation program are...

Quick Test (Cont'd)

- How can you tell if you have PCL or PostScript?
  - Ummm… throw it at the printer and if it works, it was PostScript
- GhostScript can be modified to disregard the various PCL sequences. This makes life much easier when trying to preview files produced on MS/Apple based programs

Portable Document Format (PDF)

- PDF is basically PostScript's version of PCL
  - It is very simple
  - All the time consuming PostScript operations are disallowed
  - There are some minor HyperText things thrown in to provide some previewer help
- Concept was to generate your document in PostScript or PCL, run it through a PDF converter, and you get PDF.
- To print, you expand the PDF into the more verbose PostScript, add the Job Structuring Conventions, and you are done
- Adobe now sells PDF viewers and translators...

Printing PDF

- Very few (none?) printers will accept PDF files
- You need to preprocess them into PostScript
  - Ghostscript should be able to do this
  - Whoops its cookies on many PDF files
- You can extend GhostScript to handle PDF files now with reasonable success, but you will be violating some patents, restrictions on encryption, etc etc.
- Look at the GhostScript web site for details

Portable Job Language

- Now printers have to support PCL, PostScript, TEXT, and who knows what.
- Documents need to be printed using different papers, formats, orientations, etc....
- Need a higher level language to control this type of operation, overriding (perhaps) the operations in the document
- Portable Job Language was intended to do this
- Most important feature is

  ^]-012345XPJL EOF

  – This causes a functional reset of the printer
  – Cannot be ignored, escaped, hidden, etc....
  – Restores sanity to the printing world

PJL Features

- Provides ways to specify the Page Description Language
  - PJL Select Postscript
- Provides ways to specify the orientation (if the PDL does not override it)
- Basically, provides a way of overriding the PDL requests
- And most important:
  - Provides a **standard set of error messages** to be returned IRREGARDLESS of the PCL
- This last feature is worth the pain and effort of PJL

PJL Disadvantages

- Not all PJL features are supported in all printers
- Printer vendors are very closed mouthed about what they support. There is no "PJL Printer Document" standard for PJL
- HP does not provide details on all of the error messages, leaving it up to implementers to discover that there are a whole new set of messages concerned with various printer operations
  - Try getting a paper jam in a multi-bin feed printer and see what messages you get
- Different releases of printer EPROMs support different PJL sets
  - HPXXSi are notoriously different from other HPs
- And of course - different vendors have different messages

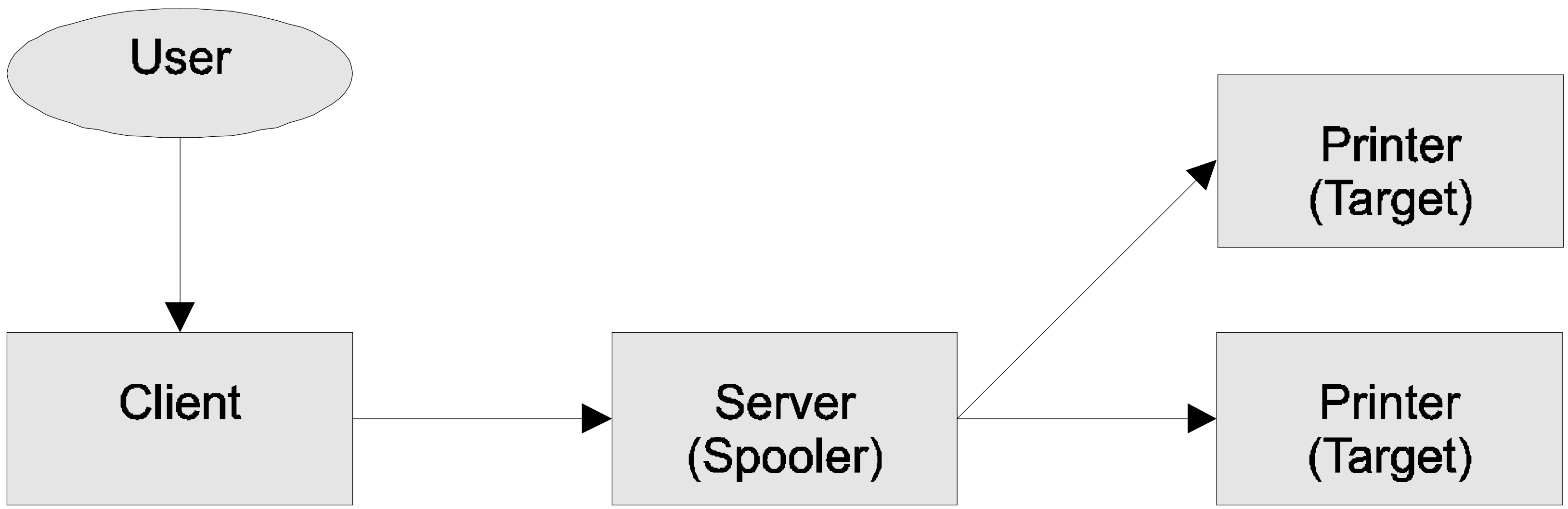
PPD Meets PJL

- If a printer supports PostScript, there is usually a PPD file for the printer.
- HP has very nicely put in the various PJL and PostScript sequences needed to perform the various printer control functions in their distributed PPD files.
- This information is not documented of course… but handy to know

Part 2

Print Spoolers

Print Spooler Basics

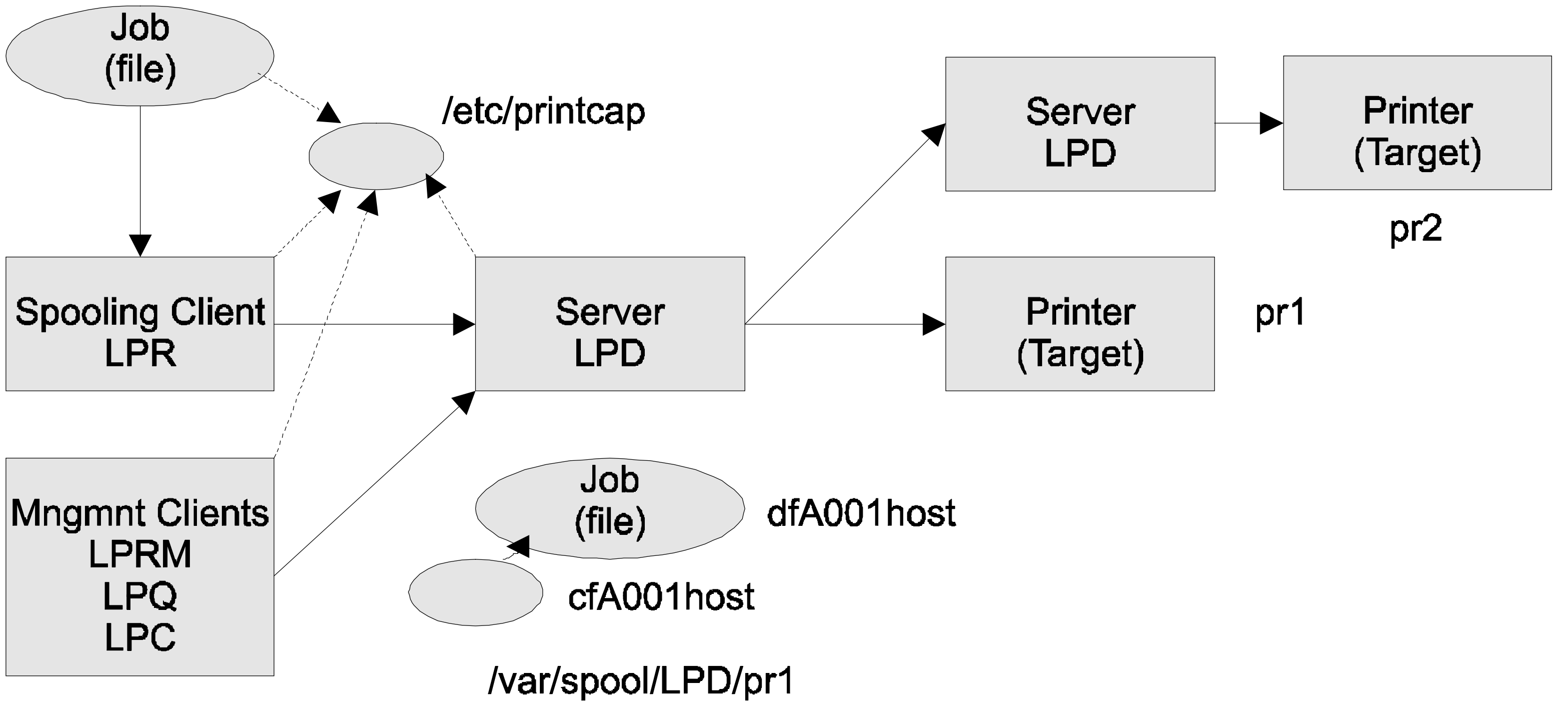


- Users create jobs (print files)
- They use a print client to send job to a print server or spooler
- The spooler then transfers jobs to a printer
- Multiple users (clients) can transfer jobs to a server
- A server can transfer multiple jobs to printers

Common Print Spooler Architectures

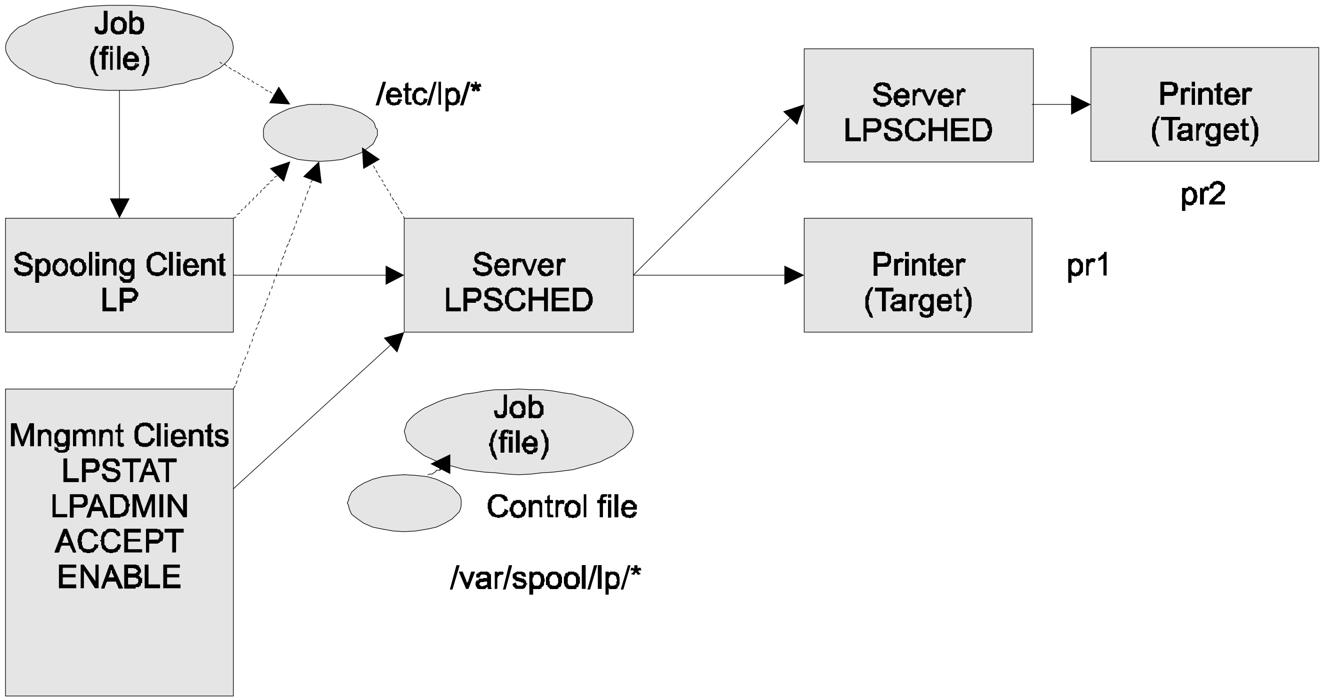
- LPR/BSD UNIX
  - variants include PLP, LPRng
  - RFC1179 documents client/server protocol
  - TCP/IP network based
- LP/SVR4 UNIX
  - Proprietary client/server protocol
- Novell Netware
  - Semi-proprietary protocol
  - IPX Network based (or IPX over TCP)
- Microsoft SMB
  - Basics documented, but details are not
  - NetBUI, IPX based

LPR (BSD/RFC1179) Architecture



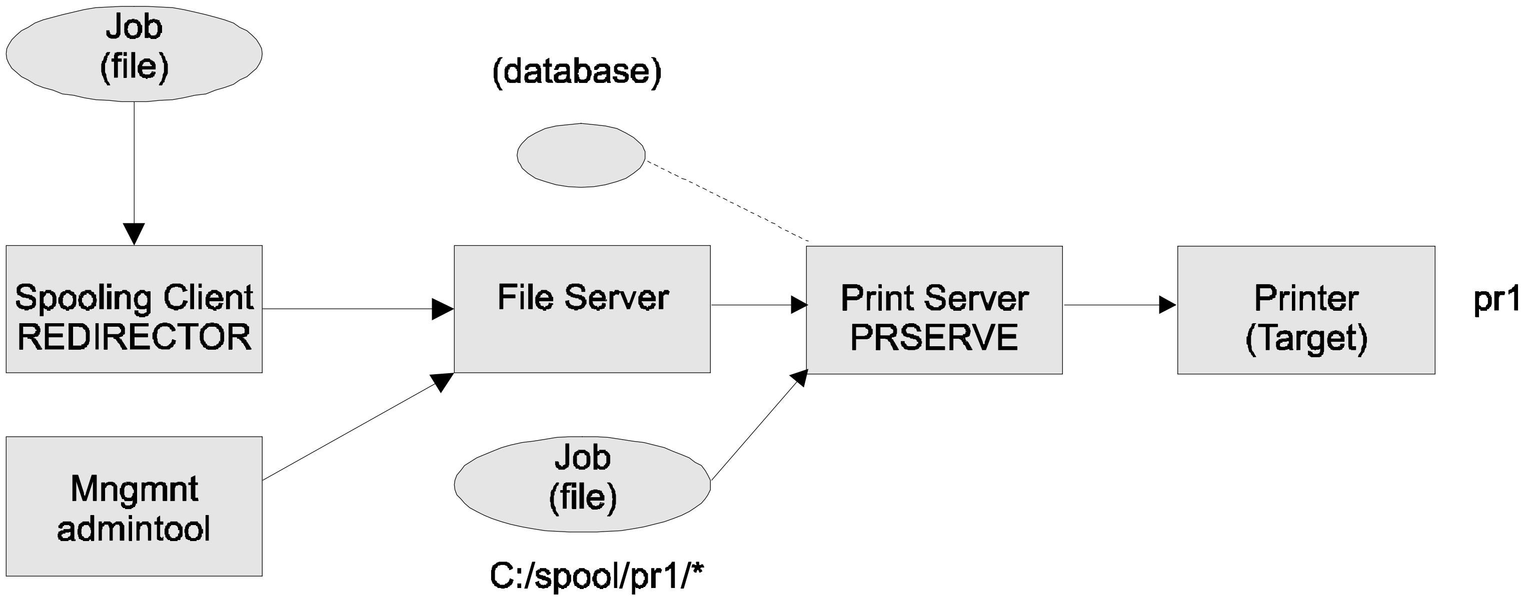
- lpd is the print daemon
  - listens on port TCP port 515 for requests from lpr clients
  - uses printcap for configuration
- lprm (job removal), lpq (job status) communicate over port 515
- jobs stored in spool queue on server host

LP (SVR4) Architecture



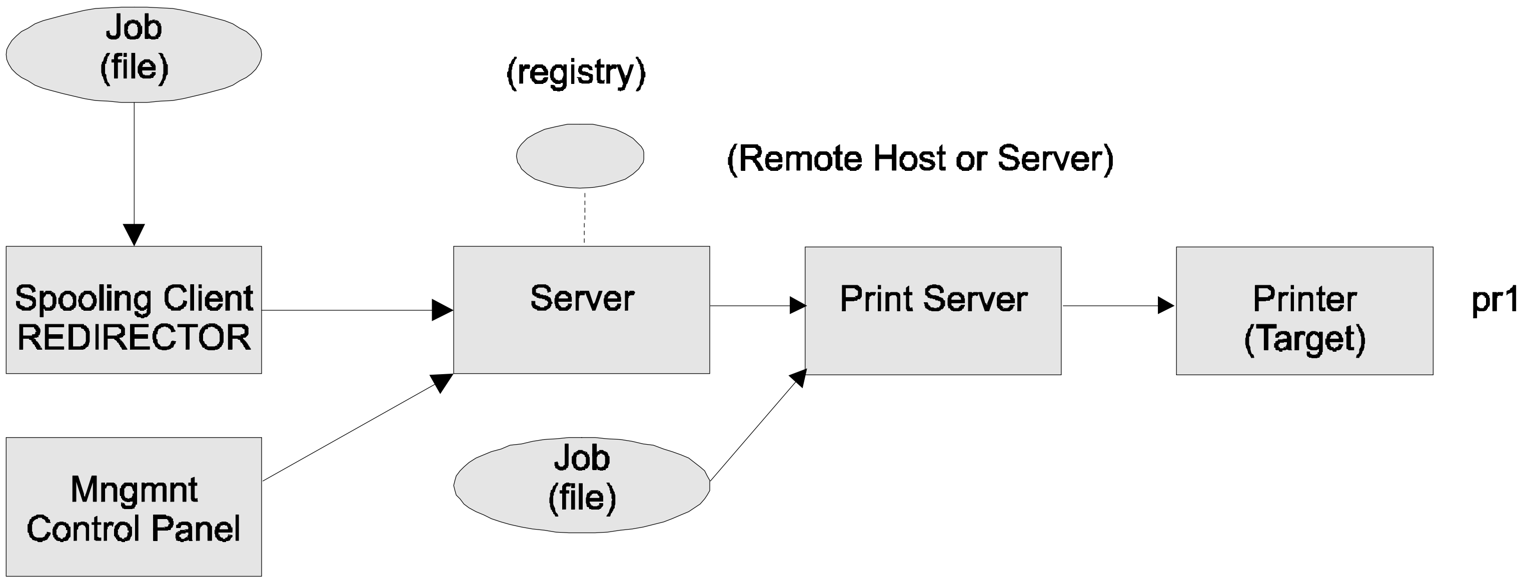
- lpsched is the print daemon
  - listens on /dev/printer (FIFO)
  - uses /etc.../lp/* files for configuration
- lpstat, lpadmin, accept, enable used to control operation
- jobs stored in spool queue on server host

Novell Print Server Architecture



- File Server is the print daemon
  - files placed on server and requests made for printing from clients
  - uses database for configuration
- printing done by server process on file server or other host
- management done using admintool or other facility
- jobs stored in spool queue on file server

NT Print Server Architecture



- Server is the print daemon
  - files placed on server and requests made for printing from clients
  - uses registry for configuration
- printing done by server process on file server or other host
- management done using control panel
- jobs stored in spool queue on file server

Observation

- Architectures are almost identical
- Should be easy to understand, right?
  - WRONG
- You can set up the various pieces easily, but the problems start when you want to do more than just fling files at printers
  - Accounting
  - Restricting access
  - Error logging and recovery

RFC 1179

Printer Protocol

RFC1179

- Documented the original BSD print spooler network protocol
- Incomplete, inconsistent, and open to abuse… I mean *implementation inconsistencies*
- Only common, non-proprietary, open standard available today
  - This situation may change as the IETF has a Internet Printing Protocol (IPP) working group making progress towards a new and sensible standard

Basic Concepts

- LPD server listens on TCP/IP port 515 for connections from client programs (LPR, LPQ, LPRM, LPC) and other LPD servers
- Connections originate from port 721 - 731 to server (Privileged Port in old TCP/IP network software).
- Clients send requests, get confirmation and/or status in return
- Request can be:
  – (LPR) transfer job
  – (LPQ) get queue status
  – (LPRM) remove job
  – (LPC LPRng Extensions) queue control

Print Job

- control file
  - contains information about the job submitter and the way the job is to be processed by the print server
  - names the data files for the job
- one or more data files
- control file and data files are transferred in binary form from client to server
  - server must interpret contents of files

Print Job Files

- File names have defined format
  - cf X nnn hostname - control file

`cfA001patrick cfA002patrick.astart.com`

  - X is a letter indicating job priority
  - nnn is a ‘job sequence number’
  - hostname is the name of the host originating the job
- Data file names should have same format
  - df X nnn hostname

`dfA001patrick dfB001patrick`

  - the X is a sequence identifier
- Order that files should be sent in is not defined
- Most network printers ignore the control files and just print data files, treating each as an independent job

Control File Format

- ASCII printable characters, line ending with \n
- Example:

`Hastart4.astart.com`	Host name
`Proot`	User name (banner)
`J(stdin)`	Job title
`CA`	Job class
`Lroot`	User name (billing)
`fdfA458astart4.astart.com`	Data file
`N(stdin)`	Data file name
`UdfA458astart4.astart.com`	Unlink data file

- Lines starting with upper case letters are information
- Lines starting with lower case letters are data files

Data File Format

- Each data file in the control file is identified by a line starting with a lower case letter.
- This letter indicates the “format” of the data file and is a hint to the server on how it should be printed.
- The **U** lines in the control file were originally used to indicate that the spooler should remove the data files after printing. By default, most spoolers do this by default.

Control File Botches

- As you see, the format of the control files is trivial
- It is amazing that so many implementations get it wrong…
  - Use non-ASCII characters (UNICODE)
  - Use CR/LF as end of line indication
  - Exceed maximum line lengths
  - Give each job the same control file/data file name
  - Give data files names like

  dfA371jobs.dat - based on name of file

Job Transfer Protocol

- Client sends a line of the form:
  `\002printer\n`
- Server responds with **`\000`**
- Client sends control file transfer request
  `\002cfXnnnhostname length\n`
- Server responds with **`\000`**
- Client sends length bytes of control file, then **`\000`**
- Server responds with **`\000`**
- Client sends data file transfer request
  `\003dfXnnnhostname length\n`
- Server responds with **`\000`**
- Client sends length bytes of data file, then **`\000`**
  `Server responds with \000`
- Repeat sending data files until all done

Protocol Headaches

- Some clients decide to send data files first, then control file
- Some clients do not send data files in same order as listed in control file
- Some clients send **`\n\r`** or **`\r\n`** instead of **`\n`**
- Some clients put non-ASCII characters in the control file
- Some clients do not use correct names for data files
- Some clients put in non-present data files
  - Now, nobody would put in a line like:

  `f/etc.../password` or **`U/etc.../password`**
  in the control file, would they? Hmm...

Printer Status (LPQ)

- RFC1179 uses the following protocol to get printer status
- Client sends

`\003printername [keys]\n`	short? format
`\004printername [keys]\n`	alternate?
`\009printername [keys]\n`	LPRng verbose

- Server responds with status, then closes the connection
- There is no definition of what the status format must be. Every LPD server returns a different one
- Keys are used to refine the status, e.g. - select a job
- No definition of what information is searched for...

Remove Job (LPRM)

- RFC1179 uses the following protocol to remove printer job
- Client sends

```
\005printername user[keys]\n
```

- Server responds with status, then closes the connection
- There is no definition of what the returned status format must be. Every LPD server returns a different one
- Keys are used to select a job
- No definition of what information is searched for...

Start Printer (LPC)

- Amazingly, RFC1179 only has one command to control a printer.
- Client sends

`\001printername\n`

- Server responds with **`\000`** and starts the printer

LPC (Not Defined by RFC1179)

- In addition to the standard functions defined by RFC1179, there is need for some sort of administration control.
- This is not part of RFC1179, and is usually implemented by the LPC program.
- Thus, there is no way to remotely manage a printer using RFC1179, leading to the use of SNMP...

BSD Print Spoolers

Why LPD/BSD?

- Uses RFC1179
- All other protocols are proprietary
- It is trivial to implement over a network, and allows any TCP/IP based protocol to provide print services
- Other protocols can be *gatewayed* to RFC1179 based printers/spoolers with very little effort
- Least common denominator in multiprotocol printing
- Available on all UNIX, Microsoft, Apple, etc... etc... platforms

Why LPRng?

- LPRng is a descendant of the LPD/BSD family of spoolers
- It provides administrative control over printing operations
- It has incredible flexibility, logging, debugging
- You pay for this by not having a simple plug and play system for non-trivial setups
  - You need to RTFM quite a bit
- From the users viewpoint, LPRng strongly resembles the LPR/BSD print spoolers
- The architecture is similar, but not identical to original BSD

LP and LPSTAT Emulation

- LPRng simulates a large subset of the SVR4 LP and LPSTAT command functionality
- You can fine tune this emulation to be more vendor specific

Network Based Administration

- Administration of printers and print queues is a major problem in large system administration is management of print queues
- On most SVR4 and BSD print systems you must log in as root, execute multiple different commands, and perhaps even delete or edit files by hand
- LPRng extends RFC1179 and provides a LPC command as well as the LPQ, LPRM, and LPR support
- There is even a strongly authenticated version using Kerberos, PGP, or SSL available (compile time option).

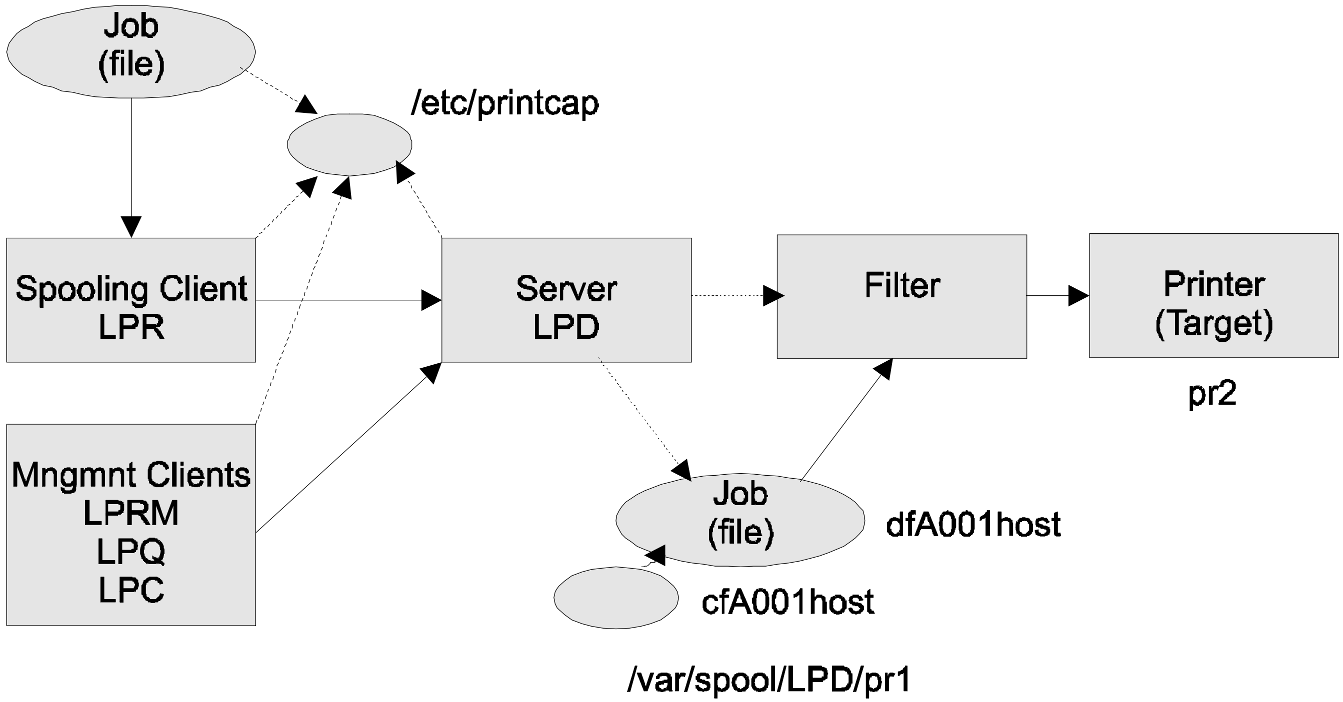
LPRng Security

- LPRng eliminates many of the security loopholes present in the original BSD code and design
- Many vendors have shipped LPD distributions with various security problems
- Remember - LPR/BSD clients run SUID root, allowing users to play games with stack overflows, etc
- LPR/BSD runs filters as root, and some filters are shell scripts with such things as

```
exec $*
```

  – Exploiting this to gain root permissions is left as an exercise for the student

LPD/BSD Details



Spooler Operation

- lpd is the print daemon
  - listens on port TCP port 515 for requests from lpr clients
  - uses printcap for configuration
- jobs stored in spool queue on server host as
  - control file with user information and list of data files to be printed
  - data files containing information to be printed
- LPD/BSD requires LPD server to run on same host as client
- LPRng allows server to be on different host

/etc/printcap

- Database used to control printer operations
- Based on the termcap format,

```
#parallel attached DUMB printer
   pr1|dumb:\
        :lp=/dev/lp:sh@:sb:mx#1000:\
        :sd=/var/spool/lpd:\
        :of=/usr/libexec/of:\
        :if=/usr/libexec/if
```

- The first part of the printcap is the primary (reference) name and the printer aliases.
- Following entries are either keywords and values, flags (**`:sb:`** set **`sb`** flag on, **`:sh@:`** sets **`sh`** flag off), or numerical values (**`:mx#1000:`** or **`:fx#0x13:`**)

Important keywords

- lp = the local printing device
- rm=remote host, rp=remote printer
  - used when jobs are to be forwarded to another host using RFC1179 job transfer
- sd=spool directory
  - where the jobs are stored
- sh - suppress (no) headers (banners) when printing locally
- sb - short (1 line) banner instead of long one
- mx # - maximum job size (0 is unlimited)

Lightweight printcap files

- LPRng eliminates much of the overhead of the BSD printcap files

```
#simple printer entry
lp|pr:lp=pr@host
```

- Client programs only need to know printer name and host running LPD server

Simplified Format

```
Lp2:server
   :sd=/var/lpd/lp2
   :lp=lp2.astart.com%9100
   :if=/usr/libexec/hpif
   :of=/usr/libexec/hpof
```

- No \ at ends of lines
- Tags can be more than 2 characters
- lp now can specify remote printer and host, as well as port
- The form **`host%port`** opens a connection to a port on the remote device - allowing direct access to printer (for PJL status reporting)
- **`server`** flags entry as used only by lpd server

New Printcap Guidlines

- Put connection or other information used by all LPRng programs in global printcap entry
- Put server only information in printcap entry AFTER the general one

```
lp:lp=lp@lprng.astart.com
# lpd only
lp:lp=/dev/lp
  :sd=/usr/local/spool/lp
  ...
```

LPR Client

- Takes a list of files, or input from STDIN, and generates a control file, transfers the control file and data files to the LPD server
- WARNING
  – BSD/LPR used to write control and data files to the spool directory, requiring SUID ROOT permissions
- LPRng uses a network connection, and takes extreme precautions when reading files; by default, LPRng's LPR runs as a user program
- The dreaded LPR -r (remove after printing) is present, but has been tamed

Formats

- When a file is submitted for printing, LPR makes a copy of the file and sends it to the LPD server
- Before printing the file, additional processing on the file may be needed
  - GIF file may need to be rasterized
- Different types of files may need different types of processing
- The processing is specified with a *format* indicator
- LPR default format is f (text **`f`**ormat?)
- LPRng allows explicit format specification

`lpr -F x`

Formats and Filters

- Filters are used to process files before sending to the printer
- Filters are specified in the printcap file as **`xf`** entries where x is the format

```
if=/usr/local/lib/filter/ifhp
vf=/usr/local/lib/filter/ifhp
```

- Some printers require some form of initialization to be done at the start and end of a job
- The of filter is used to do this processing as well as process banners or job separators generated by the lpd server
- More on filters later

Binary Files

- Some files do not require any modification before being sent to the printer (binary files)
- The **l**(**l**iteral) format is used to indicate such files
- Too many people confused 1 and **l** so LPR uses -b (binary) to specify literal format

```
lpr -b /tmp/binaryfile
```

- Just to make life miserable, some PC based LPR clients decided to use **v** format in the control file for binary files… Sigh…
- The **if** filter is used to process literal files, but is invoked with a **-c** option

```
/usr/local/filter/ifhp -c
```

Destination Printer Specification

- The LPR -Ppr option explicitly specifies the destination printer
- If not specified, the PRINTER environment variable sets the default printer; if there is no PRINTER environment variable, the first one in the printcap file is used
- Note that in LPD/BSD that the LPR client always transfers the job to the local LPD server, which stores it in the spool queue
- This led to the horrible “set symbolic link” and “remove after printing” options which have been exploited in the past to do horrible things (but not by your users, right?).

Lightweight Clients

- The standard BSD implementation requires the printing clients to transfer jobs to a LPD server running on the local host. The local LPD server then will transfer jobs to remote hosts
- LPRng implements lightweight clients, which will simply transfer jobs directly to remote hosts, eliminating the need for a server running on the local host. This also reduces file space requirements on the local host.
- If the destination is a printer that implements RFC1179, then you never have to store the print files

LPRng Printer Name Conventions

- lpr -Ppr@host
  - printcap file is not searched, network connection and default values are used
- lpr -Ppr
  - printcap file searched for entry
- lpr
  - PRINTER environment variable used as printer name
  - if no PRINTER environment variable, use first entry in printcap
  - if no printcap, use default printer value

Extensions

- If the printcap lp entry is lp=pr@host, this corresponds to or :rm=host:rp=pr:
  - use RFC1179 protocol to transfer files
- For total abuse, you can use

  `lpr -Phost%port/direct`
- This opens a TCP/IP connection to the remote port on the host and transfers the input files directly
- You really should use netcat if you want to do this:

  `nc -d host -p port files`

LPR Client and Filters

- By default, LPR simply copies files to the LPD server
- The **`lpr_filter`** option requests LPR to run the filters on the files before sending to server
- Allows localized processing and system depending hacks to be done to jobs before sending them to the spooler
- Very handy when you have vintage software whose output needs to be massaged, and uses hardwired paths to executables

Filters and Job Processing

- When a job is selected for printing, the LPD server examines the control file for data file format information

`fdfA001astart4` format f

- The printcap information is checked for a format f filter program
- The data file is piped into the filter program and the output is then sent to the printer device
- If the printer is connected via a simple TCP/IP connection to a port, I.e. - HP Jetdirect port 9100, then LPD/BSD requires special filters to open connection
- LPRng does this using lp=host%port, simplifying operation and filters tremendously

Basic Filters - Text Transformations

- A filter is given a print file on STDIN and produces output for the print device on STDOUT
- The most basic filter action is to translate LF to CF/LF combinations; most antique (vintage) printers require this
- Then we need to expand tabs
- Then we better look for sequences of \b (backspaces) and overstrikes, and replace them with the right control sequences

LPRng's **`lpf`** filter

- Extremely simple filter that will do LF to CF/LF expansion, tab expansion, and most simple printing operations such as inserting NULLS after page eject
  - NULLS? yes, sometimes you need to do this to pause long enough for the printer to do a form feed… Sigh
- When used as the OF filter, will expand a short banner string (single line) into a very nice full page banner
- This type of thing is used less and less these days, but periodically there are requests for this antique.

PostScript Printer Filters

- Many times PostScript printers are used to print text files; the IF filter should detect a non-PostScript file, and invoke a Text to PostScript translator
- Some printers stack paper in back to front order; the IF filter should try to do page reversal
- If the destination printer has PJL support, you better insert the various PJL/PCL strings to reset the printer and put it into PostScript mode.
- Did I mention getting page counts? And accounting? and storing this in an accounting file? And checking that the person has permission to print?

LPRng's `psfilter`

- This filter tries to handle most common PostScript printer problems
  - it sends reset sequences
  - it gets status information and produces error messages
  - it even logs messages returned during job printing, such as the PostScript emulators error messages
  - it gets the value of the page counter from the printer
  - It is incredibly paranoid about file formats and tries really hard to make sure that the job being sent is PostScript
  - Yes, it supports Tagged Binary Communications Protocol

APSFILTER

- ftp://sunsite.unc.edu /pub/Linux/system/printing/aps-491.tgz
- author Andreas Klemm andreas@knobel.gun.de
- co-author Thomas Bueschgens sledge@hammer.oche.de
- Very nice package that valiantly tries to determine the format of the input files (uses UNIX file utility) and then passes them through the appropriate filters.
- One of the nice features of this package is that it will even use GhostScript to produce output for a non-PostScript printer
- Strongly recommended for those desperate situations where run time and file space is no object but “User Proofing” is

LPRng APSFILTER

- Shameless copy of original APSFILTER, but done in Perl
- Closes some minor security loopholes and deals with error conditions better
- Available on the LPRng web site

LPRng's `ifhp` Filter

- Designed to handle HP printers (PCL/PJL)
- Totally shameless rip-off of the very good JetAdmin filter set produced for Solaris/SunOS
- The wide variety of HP printers makes it difficult to have a single filter automatically handle all situations, but the ifhp filter tries hard
- Various flags and options allow it to handle all known HP printers that have published PPD files with PJL information (I hope!)
- Switches between PCL and Postscript, has a text to PostScript converter

HP Printer Problems

- Different implementations of printer firmware have different bugs
- Strange (to HP) combinations of duplex, reverse, and landscape can cause catestrophic failure and require power up initialization
- Power save mode sometimes requires physical intervention when it turns on
- TCP/IP connections are left open and do not terminate correctly - you cannot connect to printer and need to power up
- Other manufacturers have equally nasty problems

Simple Parallel Port Printer

```
# parallel port printer, no banner
lp:sh:mx#0:sd=/var/spool/lpd/lp
  lp=/dev/pr
  if=/usr/libexec/lpf
```

- The sh suppresses banners, mx#0 allows unlimited size files. The printer device (**`/dev/pr`**) is opened write only by default.
- The lpf filter will do LF to CF/LF translation, expand tabs, etc....
- This is about as simple as you can make a printcap entry

PS, PJL, or PCL Parallel Port Printer

```
# parallel port printer, no banner
lp:sh:mx#0:sd=/var/spool/lpd/lp
  lp=/dev/pr
```
For PostScript printer use psfilter package
```
  if=/usr/libexec/psif -Tstatus=off
```
For PJL, PCL printer use CTI-ifhp package
```
  if=/usr/libexec/ifhp -Tstatus=off
```

- The status=off flag suppresses the filter from getting printer status, as the parallel port is write only
- The psif and ifhp filters will do Text to PostScript or PCL conversion, and will detect PostScript or PCL files
- See the psfilter and CTI-ifhp documentation for details

Simple Serial Port Printer

```
# serial port printer, no banner
lp:sh:mx#0:sd=/var/spool/lpd/lp
  :rw:lp=/dev/ttya
  :sy=9600 -echo -crmod -raw \
     -oddp -evenp pass8 cbreak ixon
  if=/usr/libexec/lpf
```

- The rw flag opens the serial port read/write
- The sy (stty) option sets the characteristics of the serial line
- Any error messages from the printer will be passed to the LPD server for action

PS, PJL, or PCL Serial Port Printer

```
# serial port printer, no banner
lp:sh:mx#0:sd=/var/spool/lpd/lp
  :rw:lp=/dev/ttya
  :sy=9600 -echo -crmod -raw \
     -oddp -evenp pass8 cbreak ixon
```

For PostScript printer use psfilter package

```
  if=/usr/libexec/psif
```

For PJL, PCL printer use CTI-ifhp package

```
  if=/usr/libexec/ifhp
```

- Since the serial port is Read/Write, the filters can query the printer for status and use this to control various operations

PS, PJL, or PCL Network Port Printer

```
# serial port printer, no banner
lp:sh:mx#0:sd=/var/spool/lpd/lp
  :rw:lp=prname%9100
```

For PostScript printer use psfilter package

```
  if=/usr/libexec/psif
```

For PJL, PCL printer use CTI-ifhp package

```
  if=/usr/libexec/ifhp
```

- The LPD server will open a connection to port 9100 on the printer, and the filters will direct their output to this port
- This is extremely effective for network based printers

Part 3

Managing Print Spooler Operations

Printers, Queues, and Status

- All jobs handled by LPD are placed in a print queue, and then sent to a printer
- Queue State
  - enabled - accepting jobs
  - disabled - not accepting jobs
- Printer (Destination) State
  - running - actively printing job
  - idle - waiting for job to print
  - stopped - administratively stopped from printing

LPQ - Status

- The LPQ command is used to request status about a printer or print queue status
- LPD/BSD has two forms of status - short and long
- LPRng adds verbose (which is REALLY verbose!)
- When a LPD server gets a status request, it replies with status for the local queue, and then if the queue is forwarded to a remote printer will forward the LPQ request.
- If the printer is local to the LPD server, the server will also report printer activity
- You can restrict the status to only selected jobs by adding keys to the request command. These usually are the user, job id, or host from which the jobs originated.

Short Status

lw4@astart4 0 jobs

Long Status

```
Printer: lw4@astart4  'Hp Laserwriter'
 Queue: no printable jobs in queue
 Status: server finished at 07:32:46
 Filter_status: ifhp Initial page count 60744,
  final 60746, Total pages = 2, elapsed time 73
  secs at Oct 29 07:32:46
 Rank   Owner/ID                          Class Job
  Files                    Size Time
error   papowell@astart4+425                A  425
  ERROR: IO error 'Broken pipe', at 10:18:48
```

Verbose Status

```
Printer: lw4@astart4
 Comment: Hp Laserwriter
 Printing: yes
 Spooling: yes
 Queue: no printable jobs in queue
 Status: printing 'papowell@astart4+576', start,
  attempt 1 at 11:28:18
 Status: opening 'astart14.astart.com' at
  11:28:18, attempt 1, timeout 10, grace 0 at
  11:28:18
 Status: accounting at start
  'papowell@astart4+576' at 11:28:18
 Filter_status: ifhp Initial page count 60744,
  final 60746, Total pages = 2, elapsed time 73
  secs at Oct 29 07:32:46
 Filter_status: ifhp Initial page count 60744,
  final 60746, Total pages = 2, elapsed time 73
  secs at Oct 29 07:32:46
```

```
Job: papowell@astart4+425 status= error
 Job: papowell@astart4+425 size= 4124
 Job: papowell@astart4+425 time= 05:58:45
 Job: papowell@astart4+425 error= IO error
   'Broken pipe', at 10:18:48
 Job: papowell@astart4+425 CONTROL=
 - Hastart4.astart.com
 - Ppapowell
 - J/tmp/a
 - CA
 - Lpapowell
 - N/tmp/a
 - fdfA425astart4.astart.com
 - UdfA425astart4.astart.com
 Job: papowell@astart4+425 HOLDFILE=
 - active_time 0
 - attempt 1
```

Status Reports

- Major weakness of print spoolers is lack of diagnostic information and job status
- LPRng assumed that most users would require information allowing them to fix paper jams, etc
- LPRng based filters generate status reports which are kept in a file in the spool queue. This information can be viewed using the lpq command
- The lpd server keeps a log file of status information as well
- This log file is also viewed using the lpq command

Alternative Status Reports

- Several CGI scripts have been developed which allow you to get LPQ status via a web browser
  - See the LPRng web site for examples
- You can write some very simple Perl Scripts that can open a connection to the LPD server and get status information.
  - See the LPRng web site for examples

LPQ and Security

- One of the major flaws in the RFC1179 protocol is that there is no way to identify the originator of a LPQ request. Thus, there is usually no restriction on the data or information returned.
- Since LPQ returns the host and user which submitted a job, this is a very good way to find the names of machines and hosts to probe when doing hacking attacks
- Firewalls should disable port 515 for this reason in both directions, as you do not want your folks hacking other systems, right?

LPRM - Job Removal

- The LPRM program generates an RFC1179 job removal request
- lprm -Plp 179 (remove job with ID 179)
- lprm -Plp john (remove FIRST job with user name john)
- lprm -Plp all (remove ALL jobs - LPRng only)
- WARNING
  - most LPD/BSD systems will accept any request from any system to delete a job.
  - The wimpy “requests must originate from port 721-731” restriction can be circumvented with a PC and a simple hacking program. Sigh…

LPC - LPD/BSD Printer Control

- The LPD/BSD system usually provides a LPC program that is used to control the LPD activities. Traditionally it does this by reading and writing files in the spool queue, and changing permissions on directories
- The details of this are too ugly for public discussion. You really don't want to know.
- This architecture means that you must log onto the server as ROOT to control the print queues

LPC - LPRng Version

- LPRng extended the RFC1179 protocol to add additional commands for LPD server administration
- The LPC program generates the appropriate request and sends it to the server
- The server will then carry out the command
- WARNING
  - The observant and suitably paranoid administrator will have noted the horrible security loophole that this has opened
  - Don't fret - we have fixed this

LPC - Basic Commands

start - enable queue for printing and start unspooling jobs

stop - disable unspooling of jobs

enable - allow jobs to be spooled

disable - prevent jobs from being spooled

status - show print spool queue status

This is simply same as LPQ, but different format

LPC - LPRng Additional Commands

abort - kill off the filters doing printing and don't restart printing

- Used when a job gets hung up, the filter is unable to process it, or there is something very strange
- Job can then be removed using LPRM

up - short for enable + start

down - short for disable + abort

restart (BSD) or kill (LPRng) - kill off the filter and then restart it

LPC - LPRng Job Control

move - move job to another print queue

```
lpc -Plp move lp2 john
```

move jobs in the lp queue to the lp2 queue which were submitted by user john

redirect - redirect all newly spooled jobs to another print queue

```
lpc -Plp redirect lp2
```

jobs submitted to the lp queue will be placed in the lp2 queue

LPC - LPRng Server Control

- **`lpc reread`**
  - the lpd server will reread the printcap and configuration information files
- **`lpc –Ppr printcap`**
  - display the printcap information that the server has for printer pr
- **`lpc –Ppr debug 1,database,receive`**
  - set real time debugging information for printer
  - lpc -Ppr debug off terminates debugging

Print Queue Job Priority and Selection

- This is explicitly undefined by RFC1179
- Most print queues run on a first come, first serve basis
- LPRng Extensions
  - **`lpr -cxray`** will give job priority X (A is lowest, Z is highest), and put it in the xray class
  - lpc class xray restricts printing to only jobs in class xray
  - lpc class off allows all classes to be printed

LPC TOPQ Command

- The LPC topq (top of queue) command will put a selected job at the top of the queue
- **`lpc –Plp topq john`** will put the first job of user john at the head of the list

Adding A Printer

Adding A Printer - Printer Checks

- Make sure printer works
  - You would be surprised at how many problems during installation are traced to non-working printers
- Check out network connection by using diagnostics or direct connection via lpr

```
lpr –Praw@ipaddr <ellipse.ps
```

- It is a good idea to run a couple of pages through the printer, power it down, and then check the page count. Some printers use an EEPROM to hold page count information, and only update it every 10 pages

Adding A Printer - Printcap Entry

- Find a printcap entry that is similar to the type of connection that you have, and append it to the printcap file.
- Run the **`checkpc`** program. This will create the necessary spool queues and files
- Spool a job to the printer:

```
lpr -Plpnew ellipse.ps
```

- Check the printing status

```
lpq -Plpnew
```

- If the job is printed correctly, then check out other formats, etc
- If it fails to print, then we move on to Diagnostics

Diagnostics and Debugging

Diagnostics

- Almost without exception, most print spoolers have horrible diagnostics
- Most of the time error reporting does not even include the time or job identification
- The software developers appear to assume that only the vendors will require detailed information or tracing information and that users or system administrators should not be aware of the internal workings or operations

LPRng Diagnostics

- LPRng is distributed as shareware or freeware
- While some limited support is provided, it is usually via email or some non-local support mechanism
- The diagnostics capability is built into the LPRng software and essentially provides an extremely verbose trace of the system operation
- Trace or log information is placed in log files, which are automatically limited to a maximum size and are truncated

Client Operation Tracing

- The same trace technique is used for client (LPR) programs
- Trace output is sent to STDERR
- Example:

  `lpr -D 1,network -Plp file 2>/tmp/log`

- General debugging level is 1, and network operation tracing is enabled

Server Operation Tracing

- For all operations except job reception and printing, trace output is sent to STDERR
- Example:

  `lpd -D1,network -F 2>/tmp/log`
- The -F flag causes the server to stay in the foreground, and allows the server to be easily terminated

Debugging Spooling and Unspooling

- When a spooling or unspooling operation is carried out by the LPD server, the printcap entry is checked for debugging flags (**`db`**) and log file (**`lf`**) entries

```
lp:sd=/usr/spool/lp:lf=log
  :db=1,receive,network:max_log#1000
```

- If the log file exists, log and trace information is appended to it
- When the log file exceeds the max_log (Kbyte) limit, it is truncated to the min_log size
- This technique allows tracing of job flows through the system in a simple and easy to follow manner

Fixing Problems

- Many of the times you will discover that LPRng balks at accepting a job from some other spooling software due to non-compliance with RFC1179 or what can be considered a security risk
- You can force LPRng to accept the job, but it will insist on 'sanitizing' it before using it
- To do this, set the fix_bad_job value in the /etc/lpd.conf file to true.

```
# fix bad control files and data file
  names
# fix_bad_job@
fix_bad_job
```

Installing
and
Setting Up
LPRng

Installing LPRng

- Get the source code from a distribution site
  - **`ftp://ftp.astart.com/pub/LPRng/`**
  - Get the distribution (**`LPRng-`**version**`.tgz`**)
  - Get the filters as well (**`LPRng_filters-`**version**`.tgz`**)
- You will need ANSI C compiler, GNU Make, GNU Zip (gzip)
- Uncompress and tar the distribution

```
gunzip -c LPRng-*.tgz |tar xf -
```

- Configure, compile, and install the code (see documentation for details

```
configure; make clean all install
```

- Install the default configuration and permissions files

```
make default
```

LPRng's `checkpc` Program

- The **`checkpc`** program reads the printcap files and checks the system spool directories for consistency with the printcap information
- When invoked by root, **`checkpc -f`** will change permissions and create the required files and directories
- It can also be used to remove old files and truncate log files.
- See the checkpc man page for details

Setting Up LPRng on BSD Systems

- You must first stop the current LPD

  `ps -axu | grep lp` get the LPD PID

  `kill` pid

- Run checkpc to fix the permissions, etc...

  `checkpc -f`

- Edit the rc.local or other rc file to start LPRng lpd

```
if [ -f /etc.../printcap ] ; then
  echo -n ' lpd' ; /usr/local/bin/lpd;
fi;
```

- Start lpd and test the system

  `lpd`

Setting Up LPRng on Solaris/SVR4

- The default printing system on Solaris is the **`lp`** print spooler. You must first disable this, as well as the various network print services

```
lpshut;
nlsadmin –r lpd tcp;
nlsadmin –r lp tcp;
```

- Check the **`/etc.../printcap`** file, then run checkpc to fix the various system files

```
checkpc –f
```

- Next, modify the rc files to start lpd instead of lpsched
- Reboot the system
- Check the system using lpq, lpr, etc....

LP System Emulation

- When the various LPRng clients are invoked using the LP system names, they will emulate the lp system behavior. This can be done by making symbolic links to the various files:

```
ln -s lpr lp;
ln -s lprm remove;
ln -s lpq lpstat;
```

- This allows programs which require the LP print system to directly use the LPR print facilities

Load Sharing
and
Printer Pools

Printer Pools and Load Sharing

- The LPRng software can do load sharing between a set of printers by having a master spool queue and a set of servers printers

Load Sharing Details

- The jobs in the master queue are sorted by priority
- When a slave printer is idle, the top priority job in the master queue will be moved to the I dleslave printer queue
- Example Printcap Entries:

```
master:ss=serv1,serv2
  :sd=/var/spool/lpd/master
serv1:sv=master
  :sd=/var/spool/lpd:lp=/dev/pr1:...
serv2:sv=master
  :sd=/var/spool/lpd:lp=/dev/pr2:…
```

- Jobs can also be spooled directly to the slave printer queues as well

Load Sharing Details (Cont'd)

- The LPD server tries to use slave printers in round robin order
- If a slave queue is disabled, no new jobs will be placed in it
- Slave printers must not perform spooling or load balancing will not work correctly

Bounce Queues

Bounce Queues and Filters

- Sometimes it is necessary to perform filtering actions on jobs and then send the job to another print queue
- This is common when dealing with network printers, or when you want to have special actions performed by a special queue

Reasons for BQ Use

```
#do 2 pages per page up
# WRONG Method
pr2up:lp=realpr@host:if=ps2up
#real printer
realpr:lp=pr%9100:sd=/sd:if=ifhp
```

- The pr2up printer simply passes 'f' format jobs forwards all job to the real printer, and does not pass the data files through the ps2up filter
- LPR will look at the pr2up entry and decide to send jobs directly to realpr, so they will never get put into the queue

Correct BQ Use

```
#do 2 pages per page up
# CORRECT Method
pr2up:lp=pr2up@host:if=ps2up
   :sd=/sd/pr2up:bq=realpr@host
#real printer
realpr:lp=pr%9100:sd=/sd/realpr:if=ifhp
```

- The pr2up printer now has an entry that will force the LPR program to send the job to the pr2up queue
- LPR now look at the pr2up entry, sees the bq entry, and will pass the data files through the ps2up filter
- The output of the filter will then be sent to the realpr queue

Alternate BQ Printcap

```
#clients see this
pr2up:lp=pr2up@host
realpr:lp=realpr@host
#lpd server adds this
pr2up:server:if=ps2up
  :sd=/sd/pr2up:bq=realpr@host
realpr:server:lp=pr%9100:sd=/sd/realpr
  :if=ifhp
```

Update Data File Formats

- After the data file has been modified, the original format may be incorrect
- Example: you want to change the original f format to l so that no further modifications are made
- The new format entry does this

```
pr2up:server:if=ps2up
  :sd=/sd/pr2up:bq=realpr@host
  :new_format=flvl
```

- New format entries are pairs of characters; the first is the original format and the second is the new format after processing
- Note that the control file is changed irregardless if there is a filter for the particular format

Editing Control Files

- A bounce queue can also be used to modify a control file. This is commonly needed when either the format of the control file is unsuitable for the destination printer, or when data file formats need to be modified as a result of filtering

```
prgif:lp=realpr@host:sd=/sd:rf=/gif2ps
  :edit_cf=/usr/lib/cf_editor
```

- The cf_editor filter is given the control file and can edit as well. Note that this can include actions such as removing job files.
- This horrible kludge is needed when dealing with vintage software that produces control files incompatible with newer network printers

Routing Jobs
to
Spool Queues

Routing

- Sometimes it is necessary to dynamically decide the spool queue to use based on information in either the control file or the format of data files
- For example: large, medium or small jobs
- For example: special color processing
- This is not the same a load sharing, which makes decisions based only on the availability of printers

Routing Filter

```
lp:lp@host
lp:server:sd=/var/spool/lp
  :router=/usr/lib/filter/router
```

- The LPD server will put jobs into the spool queue
- When unspooling them, it will pass the control file through the routing filter
- The output of the routing filter (on its STDOUT) will be the new printer name
- You can combine Bounce Queues and Routing, but the results are not predictable

Host Specific Printcap Entries

Host Specific Printcap Entry

- The **`oh`** (only for this host) entry restricts which host will use a printcap entry
- Example:

```
lp:oh=*.astart.com,130.191.163.0/24
  :lp=lp1@server
lp:oh=*.sdsu.edu,130.10.0.0/24
  :lp=lp2@server
```

- Values are lists of GLOB expressions or IP address and mask values
- If the host name or address matches, then the printcap entry can be used by the host

Part 4

Horrible Problems

Permissions
Authentication
Accounting
Defaults

Permissions

- Administrators may need to restrict access to various printing facilities for policy or financial reasons
- Most print spooler systems have some sort of mechanism for restricting access
- Problem with RFC1179 based systems is sparse information available to make decisions
  - You have the endpoint of a connection
  - You have a request type
  - Some requests have a user name, others do not

LPRng Permissions File

- Based on Packet Filter concept
- File contains a list of ACCEPT or REJECT entries
- Requests are test against entries until a match is found
- If the result is ACCEPT then the operation is allowed
- If the result is REJECT then the operation is not allowed
- You can also put in default ACCEPT or REJECT

Example1

```
# Accept LPR requests only from 10 sn
ACCEPT SERVICE=R REMOTEIP=10.0.0.0/8
REJECT SERVICE=R


# Alternative to the above
DEFAULT ACCEPT
REJECT SERVICE=R NOT REMOTEIP=10.0.0.0/10
```

Example 2

```
# Let only root and admin on server
# have LPC control permissions
ACCEPT SERVICE=C SERVER
  REMOTEUSER=root,admin
REJECT SERVICE=C
# LPRM Remove requests only from
# same host and user as spooled them
ACCEPT SERVICE=M SAMEHOST SAMEUSER
# Allow test on tester to remove files
ACCEPT SERVICE=M REMOTEUSER=test
  REMOTEHOST=tester.astart.com
```

Authentication

- RFC1179 does not provide any authentication methods
- Since it is trivial to forge network level packets, etc, this can be a major problem
- LPRng solution was to provide a general purpose method of adding authentication
- Currently, PGP and Kerberos authentication is supported

Authenticator

- A connection is established, a special AUTH request is sent
- Part of the AUTH request is the supported authentication methods
- The reply contains the chosen method to be used
- Each end of the connection starts an authenticator program which will then perform the various authentication and/or encryption to be done
- The authenticator programs accept information from the LPRng programs, encapsulate it, and then transfer it to the other end

Example: PGP

- To use PGP authentication, each LPRng server will require a secret key, and each LPRng user will need the public key of the server
- Each user will need to provide his public key to the server as well
- Each RFC1179 request and/or reply is signed with the appropriate key, identifying it as originating from the correct endpoint
- The use of a Public Key Server to supply user and/or LPRng server keys greatly simplifies this operation
- If you are really paranoid, you can also encrypt all requests and data file transfers

Forcing Authentication

- The permissions file is used to specify that some form of authentication must be done

```
ACCEPT SERVICE=R,M,C AUTH=pgp,kerberos
REJECT SERVICE=R,M,C NOT AUTH
```

Accounting

- Don’t spend dollars on counting pennies
- Levels of accounting
  - Numbers of jobs submitted
    - completed correctly?
    - Incorrectly?
  - Numbers of pages used
    - how do you find this out

Job Level Accounting

- Printcap af entry specifies the accounting file
- If the af file exists, then a message is written to it at the start and successful conclusion of each job
- Multiple starts with no conclusion indicates job failure
  - or somebody waited until next to last (blank) page came out and turned off printer…
  - Students… I mean users… become very sophisticated at this type of thing.
  - Less common today given large numbers of printers usually available for use

Page Level Accounting

- Usually required by some sort of administrative bean-counter who also counts individual paperclips and pencils
- Also, may be useful when costs need to be allocated on a project or account basis
- Most Laser based printers usually have a page counter built into them, so that you can determine the number of pages it has printed for servicing purposes
- If this pagecounter is accessible via the network interface, then you can get the value before and after a job, and determine the exact number of pages

Accessing Page Counters

- There are no standard methods to do this
  - Each model of printer appears to have a different method
  - Even documented methods may not work on printers
- High throughput printers try to do job buffering, and do not report the correct page count
  - You need to wait for TRUE END OF JOB
  - This really slows down operation
- Page Counters Lie
  - Usually value is stored in EEPROM on printer
  - You may get the stored EEPROM value, not current

Configuration and Defaults

- The **`/etc/lpd.conf`** file can be used to set values of configuration variables
- Example:

```
# fix bad control file information
fix_bad_job
```

- The file format is identical to the printcap file, but you do not need the leading colons (:) and each entry must be on a single line

Configuration Variables

- There are zillions of configuration variables used by LPRng.
- Most of these were created in order to configure LPRng operation to be compatible with very strange spooling systems or very nasty hardware
- Unless you have problems, you can safely ignore them. Really.
- If you have problems, then you better start reading the LPRng HOWTO.

Part 5

Multi-platform Printing and Gateways

Print Job Preparation

- When a print job is generated, the information must be in a format suitable for the destination printer.
- There are two general approaches to the problem:
  – Do the conversion as late as possible in the translation process (late binding)
  – Do the conversion as early as possible in the translation process (early binding)
- Late binding is favourable to the print generation program, as it does not need to know what specific printer will be used
- Early binding allows the print generation program to take advantage of any special options that are available on a particular printer

LP / LPD Uses Late Binding

- The LP and LPD/BSD print spoolers were designed to use the late binding model
- The data file format indications were supposed to indicate the type of file and the type of conversion needed
- The use of PostScript and PCL largely eliminated this need
- Most UNIX programs cannot take advantage of printer capabilities that are well known to the user

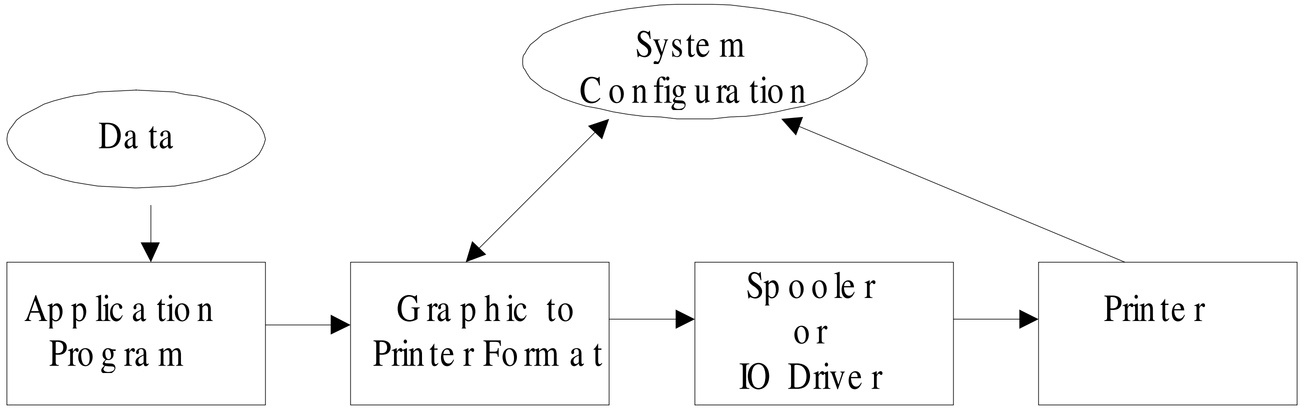
LPR -Z Options

- LPR overcomes some of these problems by allowing users to specify options that are passed to the print filters

```
lpr –Zupperbin,duplex filename
```

- The convention is that filters will scan the -Z options for values and use these values to control printing
- Since these are printer specific, the options can be different for each printer

MS Windows/MAC Use Early Binding



- System configuration information records the available printers, I.e. those which have graphic to printer format converters available to the user

Early Binding and Job Generation

- In order to print, the program needs to select a system configured printer
  - Configuration information includes the format of information to be supplied to a translation program
  - The translation program
  - Additional parameters for the translation program
- The information to be printed is processed by the translation program
- The output of the program is then sent either directly to a hardware device via another program or put in a spool file
- Spool files are then transferred either to the hardware device or a network printer

Drivers

- In the MS Windows environment, Printer drivers are actually several things
  - Graphic to print file conversion programs
  - print file to hardware interface programs
- This model leads to much confusion, as when errors occur it is difficult to determine if the problem is in the graphic to file conversion part of the driver or in the file to hardware device part of the driver
- This is even more painful when you do network printing, as now you have to distribute drivers to all printer users

Network Printing

- If you plan to print to a non-MS based print spool, choose a print driver which is as vanilla as possible
  - Try to generate PostScript or PCL
  - Try to generate PostScript Level 2 or 3 and the Document Structure Standards

Win95 and LPR/BSD Print Spoolers

- While Win95 does not directly support LPD print spooling, there are several shareware packages that provide this.

Recommended:

Windows LPR Spooler Version 4.1

Author: Susanne Heil, EDV-Vertieb

`ftp://ftp.astart.com/LPRng/wlprs41.exe`

- Implements a very good LPR/BSD RFC1179 compliant print server (LPR) and status monitor (LPQ)
- Uses Winsock interface, and works with Microsoft, FTP, and several other TCP/IP Winsock.dll

Alternatives

- There are other print spoolers available, as well as some that operate with DOS
- See the LPRng FTP site

`ftp://ftp.astart.com/LPRng/WINDOWS`

- Other utilities are there as well

Why Use Wslpr?

- If you want to have a printer pool, you will be using LPRng or some other UNIX based server (unless you have big bucks)
- It runs over TCP/IP and can go through firewalls

Why Not JetDirect?

- Be aware that using different network protocols and the JetDirect software has resulted in locking up most HP printers
- Most of the software testing was done in situations where there as little if any non-MS network activity, it appears

Windows 98

- Has support for RFC1179
- Unfortunately, it appears to have problems when connections time out and periodically jobs just vanished
- There may be a new version out Real Soon Now
- Stick with 3rd Party Spoolers

Windows NT Server LPR/BSD Gateway

- Windows NT supports remote LPR printers. It will translate print jobs spooled to an NT server by creating a control file and sends the data file
- Unfortunately, there are some minor problems with control file format, missing fields, etc., but these are easily handled by most LPD servers
- NT also ACCEPTS print jobs for printing
- This means you can use an NT server for gatewaying to the NT printer environment
- Beware that there are security holes lurking in all of this

LPRng on NT

- LPRng is being ported to NT
  - Will support the basic LPRng functionality
- Filters are very messy under NT
  - cannot easily fork processes
  - memory leaks in NT DLLs

Novell Printer LPD/BSD Gateways

- Novell also supports a LPD/BSD gateway facility
- If your Novell server has TCP/IP support installed and active, you can set up a print queue on a Novell server that will forward jobs to a LPD/BSD server.
- You can also send jobs to the Novell server and it will print them

Samba, SMB, and LPD/BSD Gateways

- If you do not want to install Wslpr on you Win95 system, or do not want to set up an NT Server, the Samba system provides an alternative
- Samba implements SMB over NetBUI and IPX
- While it is usually used for file servers, Samba also supports a very nice LPD/BSD gateway facility
- While not being personally familiar with the details of Samba, reports have indicated that it was trivial to set up and get working

"It worked first time, out of the box."

Kurt Reynolds

PCNFS and PCNFSD Gateways to LPD

- PCNFS by Sun Microsystems implements NFS (Network File System) support for PCs.
- As part of this support, Sun provides source code for the PCNFSD server which is used to authenticate PCs and provide print services via the host that the server runs on.
- The LPRng distribution has a set of patches for the PCNFSD distribution that interface the PCNFSD server to the LPRng print faciltities. They also close a couple of minor security problems involving unchecked string lengths and stack overflows

Apple and LPR/BSD

- There are several products for Apple NFS support that use the PCNFSD facilities to provide print services.
- Several people have demonstrated this working very well

LP (Solaris) to LPD Printing

- Solaris 2.5 System Administration Guide, Vol 2 Chapter 48, Example - Adding Access to a Remote Printer - **`remote@host`**

```
# define remote system as BSD, and force connection to be dropped
  when idle, 1 minute timeout between connections
lpsystem -t bsd -T 0 -R 1 host
# printer is the lp name for the print queue
lpadmin -p printer -s host!remote \
   -T unknown -I any
accept printer; enable printer
lpadmin -p printer -D "remote printer"
lpadmin -d printer # make default printer
```

Monitoring Printer Status with SNMP

- One of the weak points of distributed print spooling is monitoring printers for error conditions
- One of the benefits of network printers and the TCP/IP protocol is that you can query them for status from several different locations
- One of the nasty problems is that most printers do not provide good status indications
- The good news is that printer vendors are now implementing SNMP agents in most of their printers that allow SNMP managers to easily access printer status
- The bad news is that each vendor has different extensions and facilities

Perl, Tcl/TK to the Rescue?

- I have seen several nice printer monitoring systems developed using Perl and Tcl/TK.   Unfortunately, the most elegant ones depended on commercial SNMP managers such as OpenView and SunNetManager.
- Several interested parties have been working on a simple shareware monitor; watch the LPRng ftp site for details

Where Do I Get Help?

1. If it is a commercial system, try your vendors support group
2. comp.peripherals.printers news group has a large number of discussions about printers and spoolers
3. lprng@iona.ie mail list for LPRng related issues

   Send mail to lprng-request@iona.ie with **subscribe** in the body
4. AStArt Technologies provides commercial support for LPRng and offers network consulting and management services. (Shameless plug)

Good Luck!

Summary

- If you are going to run printers in a multiplatform environment, you will need to deal with the issues of gateways
- The LPD/BSD facilities, as defined by RFC1179, provide a common platform for use.
- While by no stretch of the imagination can this be regarded as an optimal solution, it works well in an environment with a large number of printers which need central mangement and control

Questions
and
Answers