

Fold, Spindle, and Mutilate

Printer management techniques for Macs since the paperless office has been arriving any day now for the past 30 years or so.

I'm going touch on printer settings tweaking in ways you hopefully don't know about, as well as methods for printer discovery and model identification.

I signed up for this before WWDC

About Me

- 16 years at University of Wisconsin at River Falls
- Head of macOS/iOS support
- Not the printer guy



I have worked at UW-River Falls for 16 years and am now head of macOS and iOS support. I am not charged with printer support

About UWRF

- Founded 1874, one of 13 UW Universities
- Approx 6100 students
- Approx 800 faculty and staff
- 1 Division of Technology Services
- and a few printers...

<https://www.uwrf.edu/AboutUs/FastFacts.cfm>



There are the usual workgroup laser printers,



and network copiers,



along with label printers,



large format printers,



digital presses,



printers for food labeling,



badge printers,



MICR check printers and parking ticket printers, not shown since "college is expensive" and "people sometimes park illegally" are obviously deep, dark secrets unique to UWRF.



There are also printers that make badge like labels for things that will become food.

About Printers @UWRF

- 54 Konica Minolta copiers, 236 HP LaserJets
- Almost No personal desktop printers
- PaperCut for budgeting/allocation
- Windows Print Servers/AD for printer management
- Jamf for macOS/iOS management

Most of the previously shown was or is under some form of management. The Konica Minoltas were formerly Ricoh Laniers. The 236 LaserJets is dramatically down from previously counts, and there is an even greater reduction in personal inkjets. Getting to this point was a years long project and would be its own talk in any detail.

Configuration Profiles

- Limitations in Sierra/High Sierra/Mojave:
 - No local queue name control
 - Add, but not remove
 - Very little access control
 - No settings control

Most of you have probably looked at configuration profiles for distributing or managing printers, but have been put off by their limitations. Queue names are random, you can add but not remove printers, access controls are very limited. You may be looking forward to moving past these problems in Catalina...

Configuration Profiles

- Limitations in Catalina:
 - No local queue name control
 - Add, but not remove
 - Very little access control
 - No settings control

...but then your alarm clock will go off, you'll wake up and remember that Apple has fixed none of these issues in Catalina.

Configuration Profiles

- Still too clunky to use, except
 - with agentless management
 - or, ironically, on unmanaged devices

Pure MDMs with no agent able to run scripts might still resort to printer configuration profiles, but you probably would be more interested in a better management architecture.

Unmanaged device user might appreciate installing a configuration profile over typing in an URL and selecting drivers.

Lpadmin

- Still the go to tool for printer admin work
- Changes in 10.15
 - raw is dead, unsurprisingly
 - SysV interfaces are dead
 - less fault tolerant

Lpadmin remains the best tool for printer administrator. In 10.15 using raw simply result in an error warning you not to use it. System V interfaces have been removed (the *nix greybeards in the room are crying now). In testing I notice more crashes and segfault style errors than I expected, but that may simply be because it is a beta.

lpadmin

- Still the go to tool for printer admin work
- Changes in 10.15
 - and don't panic, but...

`"PPD files and printer drivers are deprecated and will not be supported in a future version of CUPS."`

Deprecated is programmerese for dying, not dead. Unless you're using 32-bit print drivers, which should have already been a pain a few major OS versions ago, you will not need to make any changes to lpadmin calls for 10.15 .

lpadmin

- Anatomy of an lpadmin command

```
/usr/sbin/lpadmin -p localqueueName  
-D "GUI Printer Name"  
-L "Physical Location"  
-E enable  
-v lpd://servername/remotequeueName  
-P /path/to/PPD  
-o auth-info-required=negotiate  
-o whizbangoption=whizzier
```

You should all be familiar with the basic lpadmin command...

lpadmin

- but have you noticed -m?

```
/usr/sbin/lpadmin -p localqueueName  
-D "GUI Printer Name"  
-L "Physical Location"  
-E enable  
-v lpd://servername/remotequeueName  
-m modelName  
-o auth-info-required=negotiate  
-o whizbangoption=whizzier
```

The -m or Model switch uses CUPS to find a driver rather than use a PPD.

lpadmin

- `lpinfo -m`

```
drv:///sample.drv/dymo.ppd Dymo Label Printer
Library/Printers/PPDs/Contents/Resources/EPSON FAX.gz EPSON FAX
drv:///sample.drv/generpcl.ppd Generic PCL Laser Printer
drv:///sample.drv/generic.ppd Generic PostScript Printer
drv:///sample.drv/deskjet.ppd HP DeskJet Series
drv:///sample.drv/laserjet.ppd HP LaserJet Series PCL 4/5
drv:///sample.drv/intelbar.ppd IntelliBar Label Printer, 2.3
drv:///sample.drv/okidata9.ppd Oki 9-Pin Series
drv:///sample.drv/okidat24.ppd Oki 24-Pin Series
raw Raw Queue
drv:///sample.drv/zebra.ppd Zebra ZPL Label Printer
everywhere IPP Everywhere
```

You can list the built in models with `lpinfo -m`

Models may look like a keyword, URL, or similar to a POSIX filepath without a leading slash.

Most of these are too old to be useful, and the syntax is more confusing than `-P`, so why would you want to use `-m`?

lpadmin

"PPD files and printer drivers are deprecated and will not be supported in a future version of CUPS."

1) You think Apple is going to give you a choice?

For now and the coming year, your lpadmin scripts are safe, but come Snow Mountain High Catalina, there may or may not be a -P.

“Generic” Drivers

rsions/A/Resources/Airprint.ppd

2) Perhaps you've searched online for the “generic driver” path and seen an lpadmin command ending with this monstrosity...

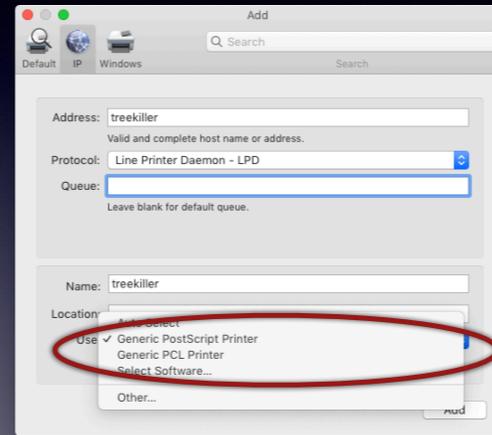
“Generic” Drivers

-m everywhere

This is how the command ends with the -m switch.

There are also some handy ways to use this with printer discovery I'll get to later.

“Generic” Drivers



Speaking of generic drivers, can anyone think of a conceptual flaw with the idea of “the” generic driver?

There are two shown in the GUI... which means there are four in macOS. One is raw, which you should never use because it is guaranteed to fail under all circumstances, so I won't talk further about it.

“Generic” Drivers

- There are three generic drivers
 - PostScript – colorful
 - PCL – compatible, greyscale, ugly
 - PDF aka AirPrint – compact, talky, the future?

PostScript is a good first choice... if you know the printer supports PostScript.

PCL is more broadly compatible, but is greyscale only and with image quality poor enough to generate complaints.

The AirPrint driver is nominally PDF based, also used by IPP Everywhere, and a good choice for any newer printer. It tends to do more calls trying to establish printer capabilities than the other generic drivers.

“Generic” Drivers

- There are three generic drivers
 - PostScript `-m drv:///sample.drv/generic.ppd`
 - PCL `-m drv:///sample.drv/generpcl.ppd`
 - PDF `-m everywhere`

Here are the model paths for them.

“Generic” Drivers

- There are three generic drivers
 - PostScript -P /System/Library/Frameworks/
ApplicationServices.framework/Versions/A/Frameworks/
PrintCore.framework/Versions/A/Resources/Generic.ppd
 - PCL -P /System/Library/Frameworks/
ApplicationServices.framework/Versions/A/Frameworks/
PrintCore.framework/Versions/A/Resources/
GenericPrinter.ppd
 - PDF -P /System/Library/Frameworks/
ApplicationServices.framework/Versions/A/Frameworks/
PrintCore.framework/Versions/A/Resources/Airprint.ppd

and the PPD paths. Don't assume these will necessarily be there as files in the future.

“Generic” Drivers

- If your printers have any capabilities of note, don't settle for generic drivers.
- When dealing with pooled or multiple output device situations, match the manufacturer and series.

Unfortunately, manufacturers don't make Mac generic drivers like those enjoyed by Windows admins.

Printer Presets

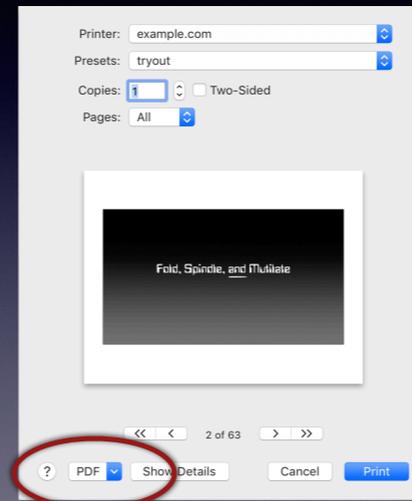
- Can apply to one printer or all.
- Stored as property list files.
- Usually in `~/Library/Preferences/com.apple.print.custompresets.forprinter.queueName.plist`

Printer Presets

- Before queueing, print jobs carry presets with them as separate files, along with additional information about the job.
- Queued print jobs have the presets encoded into them

You can take advantage of the lack of printer setting integration before spooling to speed up testing cycles while saving printer presets for distribution.

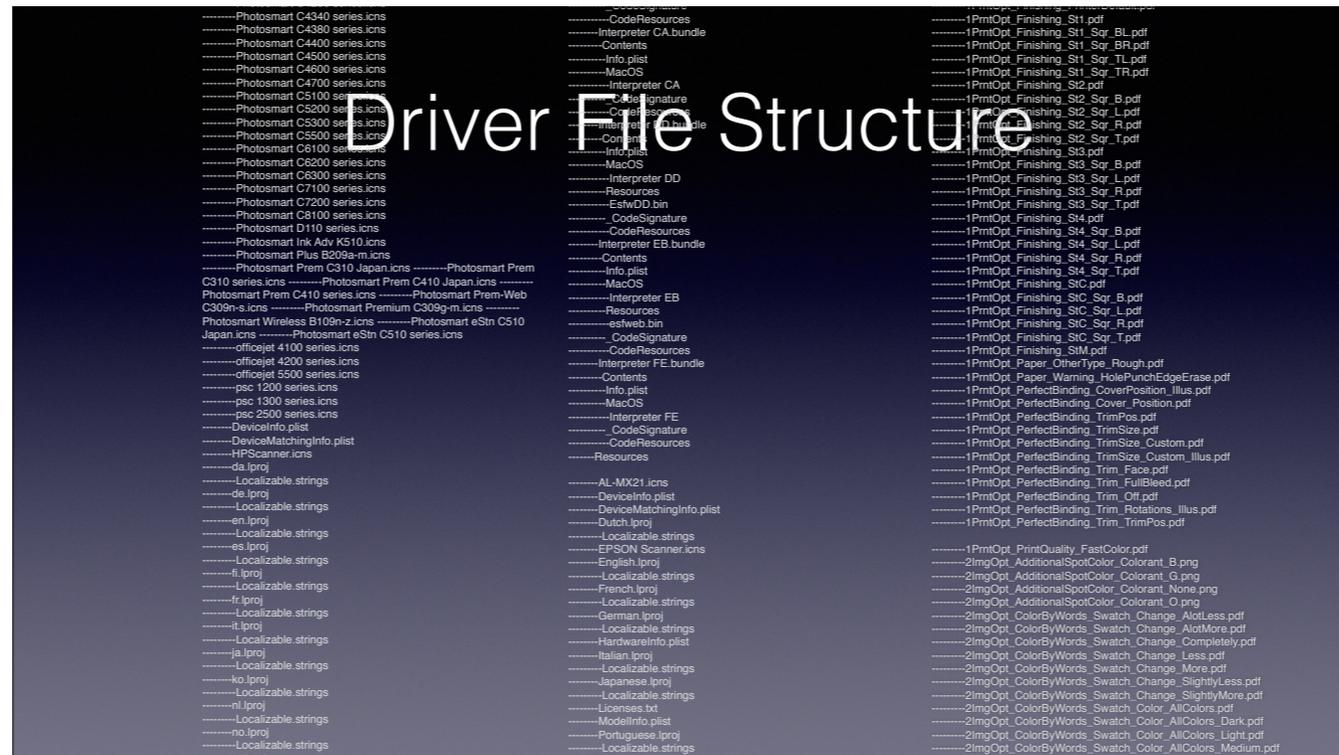
Printer Presets



>>>> Perform preset capture demo.

Paper Sizes

- Can apply to one printer or all.
- Stored as property list files.
- Usually in `~/Library/Preferences/com.apple.print.custompapers.plist`



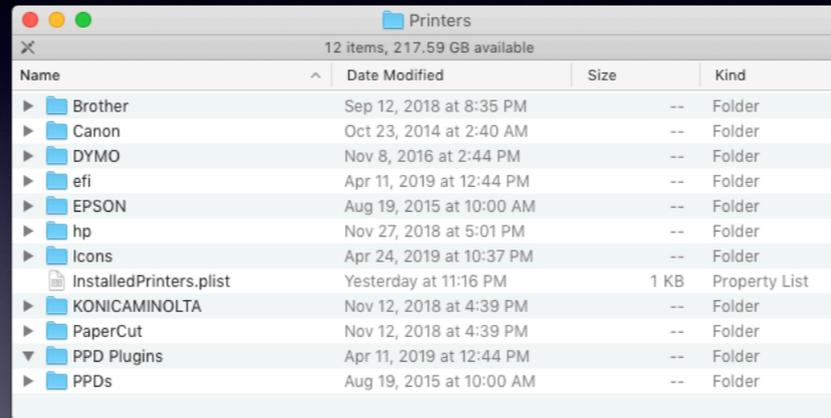
According to Pacifist the file list for the Xerox driver was about 6500 lines, HP about 6800 lines, and Epson 12400 lines. We have all day to review what these do, right?

Driver File Structure

- `/Library/Printers/`
- `/etc/cups/`

Onto driver file structures, the parts you might actually care about...

Driver File Structure



The screenshot shows a macOS Finder window titled "Printers" with a close button (X) and a status bar indicating "12 items, 217.59 GB available". The window displays a list of printer driver folders and files in a table format with columns for Name, Date Modified, Size, and Kind.

Name	Date Modified	Size	Kind
▶ Brother	Sep 12, 2018 at 8:35 PM	--	Folder
▶ Canon	Oct 23, 2014 at 2:40 AM	--	Folder
▶ DYMO	Nov 8, 2016 at 2:44 PM	--	Folder
▶ efi	Apr 11, 2019 at 12:44 PM	--	Folder
▶ EPSON	Aug 19, 2015 at 10:00 AM	--	Folder
▶ hp	Nov 27, 2018 at 5:01 PM	--	Folder
▶ Icons	Apr 24, 2019 at 10:37 PM	--	Folder
▶ InstalledPrinters.plist	Yesterday at 11:16 PM	1 KB	Property List
▶ KONICAMINOLTA	Nov 12, 2018 at 4:39 PM	--	Folder
▶ PaperCut	Nov 12, 2018 at 4:39 PM	--	Folder
▼ PPD Plugins	Apr 11, 2019 at 12:44 PM	--	Folder
▶ PPDs	Aug 19, 2015 at 10:00 AM	--	Folder

Driver File Structure

- PPDs
- Icons
- Filters & PDEs

Onto driver files, the parts you might actually care about...

PPDs

- PostScript Printer Description
- PostScript language, mostly...
- Can be a “driver” on its own, but usually bundled with icons, PDEs, filters, etc.
- Usually prepared by CUPS from a gzipped source in `/Library/Printers/PPDs/`

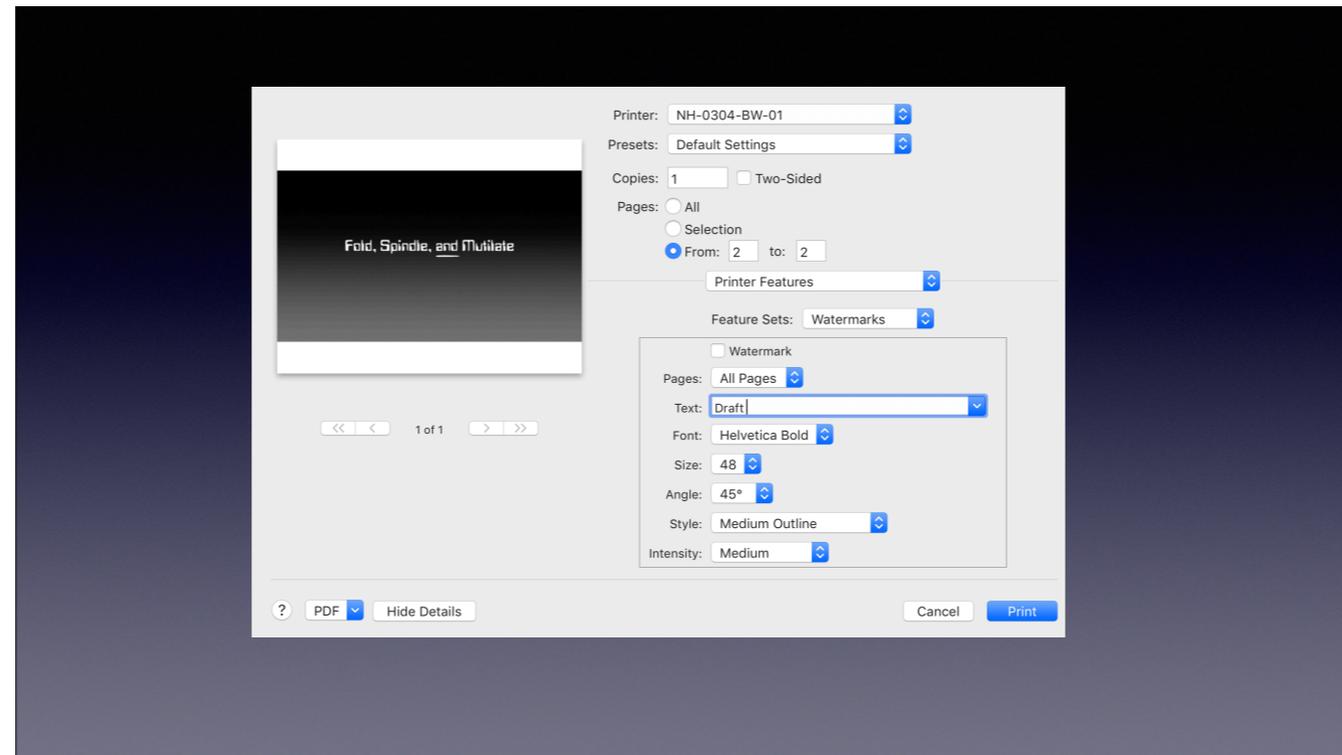
PPDs are written in PostScript, or something close but just different enough so Adobe won't sue. The PPD for a given printer tells CUPS the capabilities of the printer, and what filter (think processing program) to use to convert the print job into a format the printer can understand. If you make changes to the PPD, your users have to live with them.

PPDs

- Preferences are stored as printer presets
- Not code signed
- ASCII text, 255 character line length limit
- `cupstestppd` for a linter

Why would you want to edit these? After all, I'm sure nobody here would be the sort to void a warranty.

Changing PPDs allows you to customize features beyond what is presented in the GUI. Want to alter an icon, remove pricey to use features, or make others the default when the printer is added in a way that isn't ? Changing the CUPS prepared PPD and reissuing it is how.



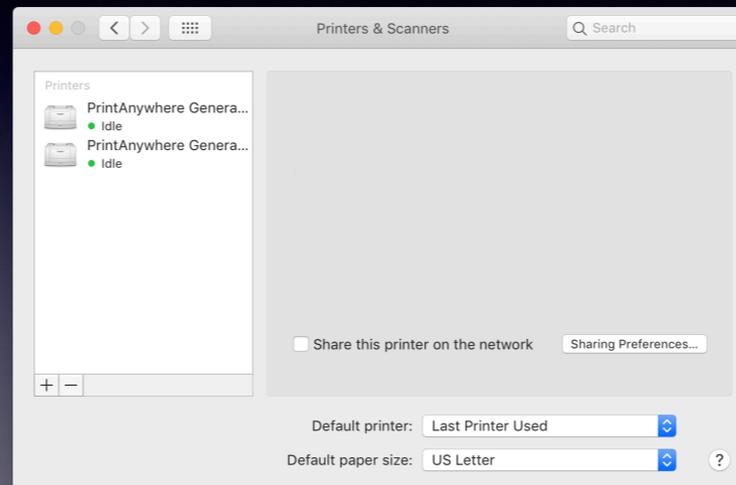
PPDs can create basic UI types like labels, pop-up buttons, text fields, check boxes, combo boxes, and radio buttons. Notably, they don't support images, or more modern types like wells, sliders, paths, and tokenized text. To get these, they used to call a Print Dialog Extension, but those are no longer available.

Icons

- Those fun, graphical thingies
- ICNS format for printer list
- Called with `APPrinterIconPath`
- Usually in `/Library/Printers/Manufacturer/Icons/`

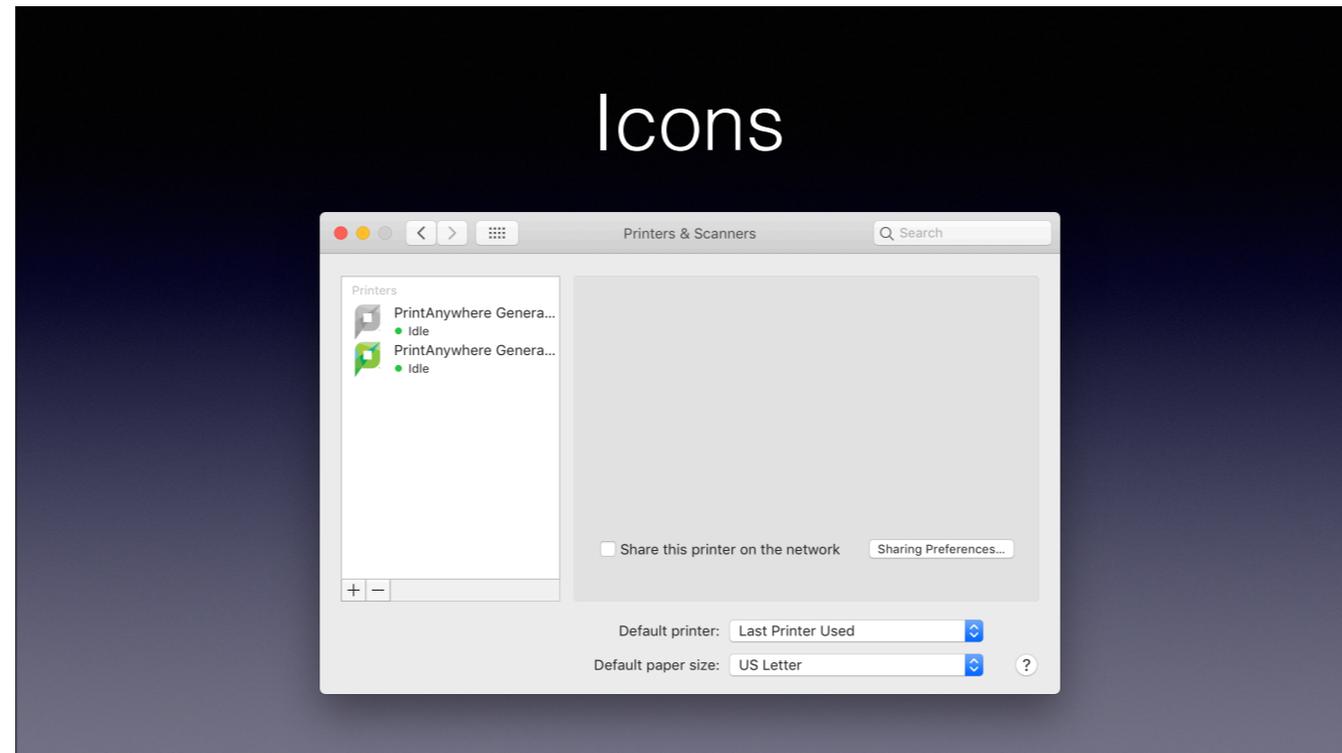
`HPPrinterIconPath` is also used by guess what brand of printer. Why would you want to change them?

Icons



Quick! Which is the color printer?

Icons



Quick! Which is the color printer?

Icons may not appear in print dialogs, but customizing them can still provide useful feedback to an inexperienced user, or help them provide feedback to you. Want to know if the user is actually using the drivers you've prepped for a printer? Customize the icon and ask them what it looks like. Got repeated instances of a printer with different features enabled, customize the icons so the user can tell them apart.

Filters

- A binary that translates the print job PostScript from the PPD to something the printer can understand.
- Not signed, but sandboxed.
- Called with `cupsFilter`

Note that the cups filter command does exhibit PostScripts unusual “arguments, then command” structure, which is not immediately obvious with other keys. Filters are of concern to admins only as part of the diagnostic process when dealing with printing problems. You can change much about them.

PPDs

```
*PPD-Adobe: "4.3"  
*%%* PPD file for LaserJet Series PCL 4/5 with CUPS.  
*%%* Created by the CUPS PPD Compiler CUPS v2.2.9.  
*% Copyright 2007-2014 by Apple Inc.  
*% Copyright 1997-2007 by Easy Software Products.  
*%  
*% These coded instructions, statements, and computer programs are the  
*% property of Apple Inc. and are protected by Federal copyright  
*% law. Distribution and use rights are outlined in the file "LICENSE.txt"  
*% which should have been included with this file. If this file is  
*% file is missing or damaged, see the license at "http://www.cups.org/".  
*FormatVersion: "4.3"  
*FileVersion: "2.1"  
*LanguageVersion: English  
*LanguageEncoding: ISOLatin1  
*PCFileName: "laserjet.ppd"  
*Product: "(LaserJet Series PCL 4/5)"  
*Manufacturer: "HP"  
*ModelName: "HP LaserJet Series PCL 4/5"  
*ShortNickName: "HP LaserJet Series PCL 4/5"  
*NickName: "HP LaserJet Series PCL 4/5"  
*PSVersion: "(3010.000) 0"  
*LanguageLevel: "3"  
*ColorDevice: False
```

Behold the start of a PPD file. This is one of those bundled with CUPS.

PPDs

```
*PPD-Adobe: "4.3"  
*PPD file for LaserJet Series PCL 4/5 with CUPS.  
*Created by the CUPS PPD Compiler CUPS v2.2.9.  
*Copyright 2007-2014 by Apple Inc.  
*Copyright 1997-2007 by Easy Software Products.  
*  
*These coded instructions, statements, and computer programs are the  
*property of Apple Inc. and are protected by Federal copyright  
*law. Distribution and use rights are outlined in the file "LICENSE.txt"  
*which should have been included with this file. If this file is  
*file is missing or damaged, see the license at "http://www.cups.org/".  
*FormatVersion: "4.3"  
*FileVersion: "2.1"  
*LanguageVersion: English  
*LanguageEncoding: ISOLatin1  
*PCFileName: "laserjet.ppd"  
*Product: "(LaserJet Series PCL 4/5)"  
*Manufacturer: "HP"  
*ModelName: "HP LaserJet Series PCL 4/5"  
*ShortNickName: "HP LaserJet Series PCL 4/5"  
*NickName: "HP LaserJet Series PCL 4/5"  
*PSVersion: "(3010.000) 0"  
*LanguageLevel: "3"  
*ColorDevice: False
```

The PPD-Adobe line acts like a shebang on a script. There are variants of it like %!PS which would still be valid.

The ModelName key is consulted to find a driver when you use the -m switch in CUPS.

PPDs

```
*modelName: "HP LaserJet Series PCL 4/5"  
*shortNickName: "HP LaserJet Series PCL 4/5"  
*nickName: "HP LaserJet Series PCL 4/5"  
*PSVersion: "(3010.000) 0"  
*LanguageLevel: "3"  
*ColorDevice: False  
*DefaultColorSpace: Gray  
*FileSystem: False  
*Throughput: "8"  
*LandscapeOrientation: Plus90  
*TRasterizer: Type42  
*% Driver-defined attributes...  
*1284DeviceID: "MFG:HP;MDL:HP LaserJet;CMD:PCL;"  
*cupsBackSide: "Normal"  
*cupsVersion: 2.2  
*cupsModelNumber: 0  
*cupsManualCopies: False  
*cupsFilter: "application/vnd.cups-raster 50 rastertohp"  
*cupsLanguages: "en ar ca cs da de el es fi fr he hr hu id ms no pl pt -"  
*UIConstraints: *Duplex *Option1 False  
*UIConstraints: *Option1 False *Duplex  
*UIConstraints: *PageSize A3 *InputSlot Envelope  
*UIConstraints: *InputSlot Envelope *PageSize A3  
*UIConstraints: *InputSlot Envelope *PageSize A3
```

The ColorDevice and DefaultsColorSpace tell CUPS if you have a color or black and white printer. You can change them to force greyscale printing, though you might want to look for manufacturer specific keys to do this.

PPDs

```
*UIConstraints: *Duplex *Option1 False
*UIConstraints: *Option1 False *Duplex
*UIConstraints: *PageSize A3 *InputSlot Envelope
*UIConstraints: *InputSlot Envelope *PageSize A3
*UIConstraints: *PageSize A4 *InputSlot Envelope
*UIConstraints: *InputSlot Envelope *PageSize A4
*UIConstraints: *PageSize A5 *InputSlot Envelope
*UIConstraints: *InputSlot Envelope *PageSize A5
*UIConstraints: *PageSize B5 *InputSlot Envelope
*UIConstraints: *InputSlot Envelope *PageSize B5
*UIConstraints: *PageSize Executive *InputSlot Envelope
*UIConstraints: *InputSlot Envelope *PageSize Executive
*UIConstraints: *PageSize Legal *InputSlot Envelope
*UIConstraints: *InputSlot Envelope *PageSize Legal
*UIConstraints: *PageSize Letter *InputSlot Envelope
*UIConstraints: *InputSlot Envelope *PageSize Letter
*UIConstraints: *PageSize Tabloid *InputSlot Envelope
*UIConstraints: *InputSlot Envelope *PageSize Tabloid
*OpenUI *PageSize/Media Size: PickOne
*OrderDependency: 10 AnySetup *PageSize
*DefaultPageSize: Letter
*PageSize Letter/US Letter: "<</PageSize[612 792]/ImagingBBox null>>setpagedevice
*PageSize Legal/US Legal: "<</PageSize[612 1008]/ImagingBBox null>>setpagedevice
*PageSize Executive/Executive: "<</PageSize[522 756]/ImagingBBox null>>setpagedevice
```

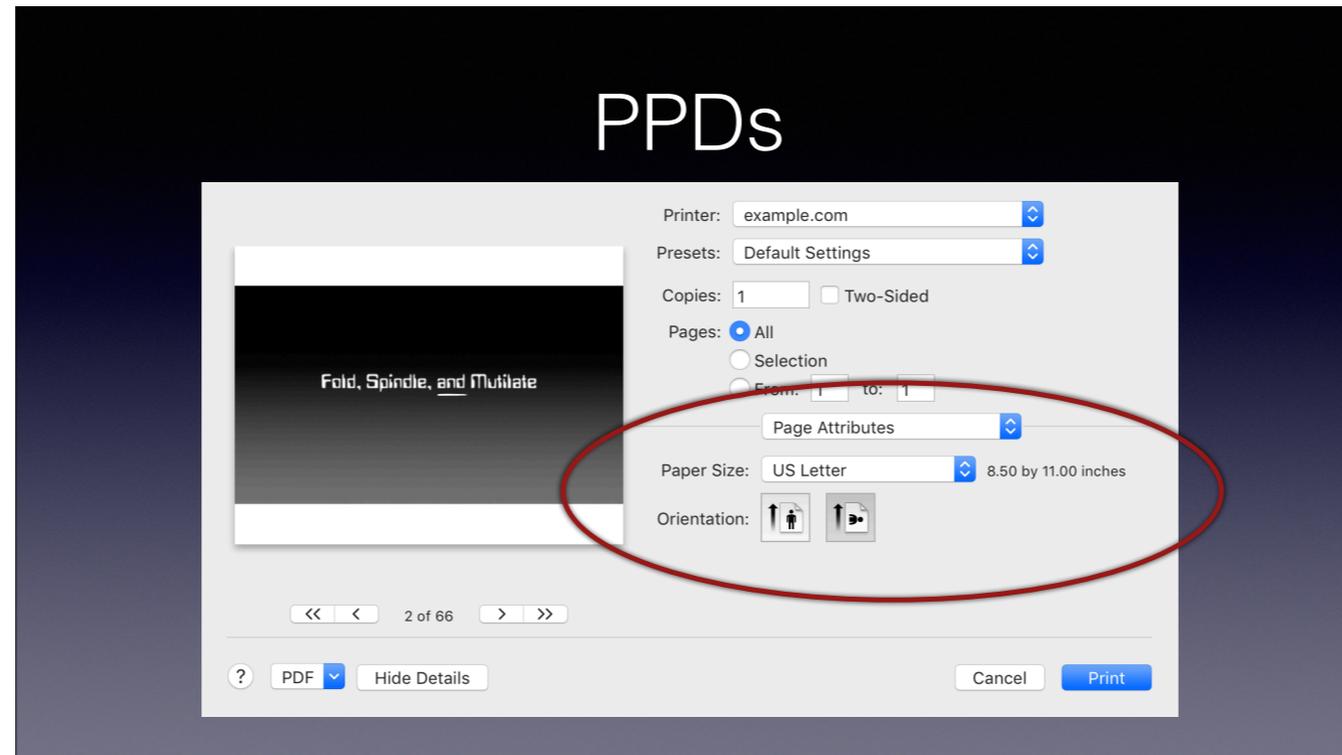
Behold UI constraints. Change the location of the last asterisk to change the default value, but still allow your users to change the default setting.

PPDs

```
*OpenUI *PageSize/Media Size: PickOne
*OrderDependency: 10 AnySetup *PageSize
*DefaultPageSize: Letter
*PageSize Letter/US Letter: "<</PageSize[612 792]/ImagingBBox null>>setpagedevice"
*PageSize Legal/US Legal: "<</PageSize[612 1008]/ImagingBBox null>>setpagedevice"
*PageSize Executive/Executive: "<</PageSize[522 756]/ImagingBBox null>>setpagedevice"
*PageSize Tabloid/Tabloid: "<</PageSize[792 1224]/ImagingBBox null>>setpagedevice"
*PageSize A3/A3: "<</PageSize[842 1191]/ImagingBBox null>>setpagedevice"
*PageSize A4/A4: "<</PageSize[595 842]/ImagingBBox null>>setpagedevice"
*PageSize A5/A5: "<</PageSize[420 595]/ImagingBBox null>>setpagedevice"
*PageSize B5/JIS B5: "<</PageSize[516 729]/ImagingBBox null>>setpagedevice"
*PageSize EnvISOB5/Envelope B5: "<</PageSize[499 709]/ImagingBBox null>>setpagedevice"
*PageSize Env10/Envelope #10: "<</PageSize[297 684]/ImagingBBox null>>setpagedevice"
*PageSize EnvC5/Envelope C5: "<</PageSize[459 649]/ImagingBBox null>>setpagedevice"
*PageSize EnvDL/Envelope DL: "<</PageSize[312 624]/ImagingBBox null>>setpagedevice"
*PageSize EnvMonarch/Envelope Monarch: "<</PageSize[279 540]/ImagingBBox null>>setpagedevice"
*CloseUI: *PageSize
*OpenUI *PageRegion/Media Size: PickOne
*OrderDependency: 10 AnySetup *PageRegion
*DefaultPageRegion: Letter
*PageRegion Letter/US Letter: "<</PageRegion[612 792]/ImagingBBox null>>setpagedevice"
```

Groups of printer preferences are always enclosed with OpenUI and CloseUI.

PPDs



The previously show lines correspond to this in the GUI.

Discovery

- Network segregation
- Private list
- DNS-SD (Airprint & IPP everywhere)
- Directory Services

Discovery

Network Segregation

- Printers are only one a selected number of subnets or VLANs
- Works if you already got it, substantial setup required.
- Coarse access control

Designing good printer management means designing good network management, if only to keep your users from bypassing access controls. However, I think of this as more a step towards printer management, not **the** method.

Discovery Network Segregation

```
AppleScript ⌵ | tell current application ⌵ | <No selected element> ⌵ |
set printserver to button returned of (display dialog "What type of network printer would you like to add?" buttons ("Office", "Lab") default button 1)
set printserver to (run script (printserver & " of {office:'139.225.51.\', lab:'139.225.159.\'}"))

if (do shell script "route -n get default") does not contain "139.225." then
    error "Not on the UWRF campus network. Check connection."
end if

set printlist to (do shell script "echo " > ~/printhosts.txt; for ((i=1;i<256;i++)); do host " & printserver & "$i | grep -v NXDOMAIN | cut -d' ' -f5 >>
~/printhosts.txt; done; sort ~/printhosts.txt")

set printchoice to choose from list (paragraphs of printlist) with prompt "Choose one or more printers:" with multiple selections allowed without empty
selection allowed

if printchoice is false then
    error "User cancelled."
end if

repeat with i in printchoice
    set printqueue to
        set printname to do shell script "echo " & quoted form of printqueue & " | tr -cd [:alnum:];"
        do shell script "lpadmin -p " & printname & " -D "" & printqueue & "" -E -v 'lpd:/' & printqueue & "" -o printer-is-shared=false -P /System/Library/
Frameworks/ApplicationServices.framework/Versions/Current/Frameworks/PrintCore.framework/Resources/Generic.ppd"
    end repeat

tell application "System Preferences"
    open ((path to system preferences) as string) & "PrintandScan.prefpane"
    activate
end tell
```

Designing good printer management means designing good network management, if only to keep your users from bypassing access controls. However, I think of this as more a step towards printer management, not **the** method.

Discovery

Private list

- You, or a management architecture, keeps a list of printers and who gets them.
- Easy, iterative setup.
- Good access control.
- Headache to maintain, scales very poorly.

The list may not take the form of a single list or database. Munki, FleetSmith, or Jamf issuing printers to groups is a private list method, even though there isn't a single list.

Discovery

- Managing printers one by one for multiple departments quickly gets unworkable
- Letting users find printers gives them a sense of control and you back some sanity.



Printers, like tablets and laptops, are physical objects that give data form in the real world. People can see, hear, and touch them, and so users are aware of them. We are all Mac admins, we are not Christopher Columbus. We do not need to discover things that our users already know about. We do not need to get into departmental catfights over who uses what printer. Let your users do the work of allocating printers to themselves and you'll both be happier.

Discovery

DNS-SD

- Domain Name Service - Service Discovery
- Technology behind AirPrint and IPP Everywhere
- Easy to implement
- Limited access control, scales poorly

Discovery

DNS-SD

- `lpinfo -v`
- `ippfind`

If for some reason you want to access dns-sd discovery programmatically, the commands are `lpinfo -v`, and parse for anything starting with “network dnssd://” , and `ippfind`, which will produce a clean list of URLs. However, carefully consider your motives for doing so in either case.

Discovery Directory Service

- ActiveDirectory and OpenDirectory
- Substantial work to implement
- Lets the users find the printers they want
- Very good access control, scales well

This is the method UWRF uses, along with a quartet of virtual print hold queues.

Directory Services

- Most will let you discover the printer but, at least for Macs, not the driver.
- Depends on the computer being bound to the directory... unless

Discovery with ActiveDirectory or the LDAP will not give you a drivers. Windows print servers don't know what to do with SNMP calls, and don't relay them.

Directory Services

SMButil

- Can query SMB servers, including Windows print servers, for available resources.
- Kerberos savvy, or use a service account
- Output is reasonably easy to parse

SMButil view will look up information on Windows server shares, including printers. If a Kerberos ticket is available, it will use it.

smbclient

• `smbutil view //printserver.example.com`

Share	Type	Comments
mcs-0111-bw-01	Printer	HP LaserJet MFP M426fdn
hh-0140-color-01	Printer	Ricoh MP C2003
uc-0170-color-02	Printer	HP Color LaserJet CP4525
hat-b003-label-01	Printer	Dymo LabelWriter Twin
cs-0151-bw-01	Printer	HP LaserJet 1320
C\$	Disk	Default share
kfa-0308-color-01	Printer	HP Color LaserJet 4700
par135-bw-01	Printer	HP LaserJet M402dne
print\$	Disk	Printer Drivers
mcm110-bw-01	Printer	HP LaserJet M402dn
rdi-0120-color-01	Printer	HP Color LaserJet CP4525
jhn-0119-bw-01	Printer	HP LaserJet P4015
IPC\$	Pipe	Remote IPC

13 shares listed

Here is the typical output of `smbutil view`. You'll notice some things in this list aren't printers. `Grep`, `sed`, or `awk` the output to eliminate those.

The comment attribute in AD corresponds to the comments column. shown. However, your results will look more blank, since I asked our printer guy to set the AD comment to the model string. Unfortunately you don't get some of the other juicy details that `dscl` will yield on a bound mac.

smbclient

```
AppleScript ⌵ tell current application ⌵ <No selected element> ⌵
set printserver to button returned of (display dialog "What type of network printer would you like to add?" buttons {"Office", "Lab"} default button 1)
set printserver to (run script (printserver & " of {office:\"officeprinter.uwr.edu\", lab:\"labprinter.uwr.edu\"}"))
set printlist to do shell script "smbutil view //ADDDOMAIN;printacct:printpw@" & printserver & "" | grep Printer | grep -v Disk | sed s/\\ Printer/\\ /g | sort"
if printlist is "" then
    error "Unable to retrieve a list of printers. Check network connection."
end if
set printchoice to choose from list (paragraphs of printlist) with prompt "Choose one or more printers:" with multiple selections allowed without empty selection allowed
if printchoice is false then
    error "User cancelled."
end if
repeat with i in printchoice
    set printqueue to text 1 thru (offset of space in i) in i
    set printname to do shell script "echo " & quoted form of printqueue & " | tr -cd [:alnum:]-"
    do shell script "lpadmin -p " & printname & " -D "" & printqueue & "" -E -v 'smb://'" & printserver & "/" & printqueue & "" -o printer-is-shared=false -P /System/Library/
    Frameworks/ApplicationServices.framework/Versions/Current/Frameworks/PrintCore.framework/Resources/Generic.ppd"
end repeat
tell application "System Preferences"
    open ((path to system preferences) as string) & "PrintandScan.prefpane"
    activate
end tell
```

lpinfo

```
lpinfo --product "$model_string" --language English -m;
```

lpinfo can search for drivers matching the model string. The driver it finds can then be used with the lpadm -m switch.

Note that it can return multiple matches, so you'll need to account for this in scripts. I usually just take the first match.

lpinfo

```
mx00000% lpinfo --product "HP Officejet Pro L7700 series" --  
language English -m;
```

```
Library/Printers/PPDs/Contents/Resources/HP Officejet Pro  
L7700.ppd.gz HP Officejet Pro L7700 series
```

```
Library/Printers/PPDs/Contents/Resources/HP Officejet Pro  
L7700 Series Fax.ppd.gz HP Officejet Pro L7700 series. Fax
```

```
everywhere IPP Everywhere
```

Split on .gz or .ppd . Note that it can return multiple matches, so you'll need to account for this in scripts.

snmpwalk

```
snmpwalk -Os -c public -v 1 someprinter.example.com
```

It will tell you more about a printer than you ever wanted to know, but requires extensive parsing and requires the **client** to have direct network access to the printer.

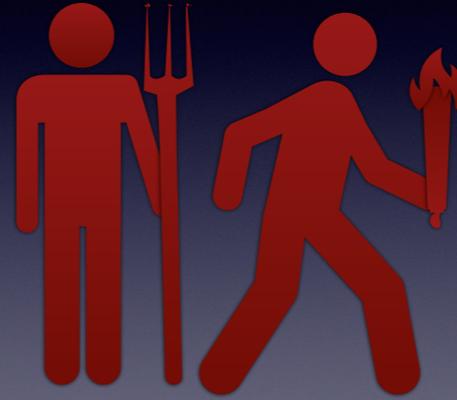
Slides & Scripts

<https://github.com/smtc474/PSUMacAdmins2019>



Questions & Feedback

- <https://bit.ly/psumac2019-333>



Printer Passwords

```
security add-internet-password -a "printpw" -D "Network  
Password" -s "labprintserver.example.com" -r "smb " -l  
"Display Name of Printer" -A -w 'haxorP@ssw0rd'
```

This slide was cut from the presentation for time, then shown during Q&A