UNIX
PRINTING
SYSTEM

CUPS Software Administrators Manual

CUPS–SAM–1.1.18

Table of Contents

Table of Contents

Table of Contents

Table of Contents

Table of Contents

Table of Contents

Preface

This software administrators manual provides printer administration information for the Common UNIX Printing System^{TM} ("CUPS^{TM}"), version 1.1.18.

System Overview

CUPS provides a portable printing layer for UNIX®−based operating systems. It has been developed by Easy Software Products to promote a standard printing solution for all UNIX vendors and users. CUPS provides the System V and Berkeley command−line interfaces.

CUPS uses the Internet Printing Protocol ("IPP") as the basis for managing print jobs and queues. The Line Printer Daemon ("LPD") Server Message Block ("SMB"), and AppSocket (a.k.a. JetDirect) protocols are also supported with reduced functionality. CUPS adds network printer browsing and PostScript Printer Description ("PPD") based printing options to support real−world printing under UNIX.

CUPS also includes a customized version of GNU Ghostscript (currently based off GNU Ghostscript 5.50) and an image file RIP that are used to support non−PostScript printers. Sample drivers for HP and EPSON printers are included that use these filters.

Document Overview

This software administrators manual is organized into the following sections:

Notation Conventions

Various font and syntax conventions are used in this guide. Examples and their meanings and uses are explained below:

Example	Description
`lpstat` `lpstat(1)`	The names of commands; the first mention of a command or function in a chapter is followed by a manual page section number.
ize /var */usr/share/cups/data/testprint.ps*	File and directory names.
`Request ID is Printer-123`	Screen output.
`lp –d printer filename ENTER`	Literal user input; special keys like **ENTER** are in ALL CAPS.
12.3	Numbers in the text are written using the period (.) to indicate the decimal point.

Document Overview

Abbreviations

The following abbreviations are used throughout this manual:

kb
> Kilobytes, or 1024 bytes

Mb
> Megabytes, or 1048576 bytes

Gb
> Gigabytes, or 1073741824 bytes

Other References

CUPS Software Programmers Manual
> A programmer guide for interfacing with and/or extending the CUPS software.

CUPS Software Users Manual
> An end−user guide for using the CUPS software.

1 – Printing System Overview

This chapter provides an overview of how the Common UNIX Printing System works.

The Printing Problem

For years *the printing problem* has plagued UNIX. Unlike Microsoft® Windows® or Mac OS, UNIX has no standard interface or system in place for supporting printers. Among the solutions currently available, the Berkeley and System V printing systems are the most prevalent.

These printing systems support line printers (text only) or PostScript printers (text and graphics), and with some coaxing they can be made to support a full range of printers and file formats. However, because each varient of the UNIX operating system uses a different printing system than the next developing printer drivers for a wide range of printers and operating systems is extremely difficult. That combined with the limited volume of customers for each UNIX varient has forced most printer vendors to give up supporting UNIX entirely.

CUPS is designed to eliminate *the printing problem*. One common printing system can be used by all UNIX varients to support the printing needs of users. Printer vendors can use its modular filter interface to develop a single driver program that supports a wide range of file formats with little or no effort. Since CUPS provides both the System V and Berkeley printing commands, users (and applications) can reap the benefits of this new technology with no changes.

The Technology

CUPS is based upon an emerging Internet standard called the Internet Printing Protocol. IPP has been embraced by dozens of printer and printer server manufacturers and is supported by Microsoft Windows 2000.

IPP defines a standard protocol for printing as well as managing print jobs and printer options like media size, resolution, and so forth. Like all IP−based protocols, IPP can be used locally or over the Internet to printers hundreds or thousands of miles away. Unlike other protocols, however, IPP also supports access control, authentication, and encryption, making it a much more capable and secure printing solution than older ones.

IPP is layered on top of the Hyper−Text Transport Protocol ("HTTP") which is the basis of web servers on the Internet. This allows users to view documentation, check status information on a printer or server, and manage their printers, classes, and jobs using their web browser.

CUPS provides a complete IPP/1.1 based printing system that provides Basic, Digest, and local certificate authentication and user, domain, or IP−based access control. TLS encryption will be available in future versions of CUPS.

Jobs

Each file or set of files that is submitted for printing is called a *job*. Jobs are identified by a unique number starting at 1 and are assigned to a particular destination, usually a printer. Jobs can also have options associated with them such as media size, number of copies, and priority.

Classes

CUPS supports collections of printers known as *classes*. Jobs sent to a class are forwarded to the first available printer in the class.

Filters

Filters allow a user or application to print many types of files without extra effort. Print jobs sent to a CUPS server are filtered before sending them to a printer. Some filters convert job files to different formats that the printer can understand. Others perform page selection and ordering tasks.

CUPS provides filters for printing many types of image files, HP−GL/2 files, PDF files, and text files. CUPS also supplies PostScript and image file Raster Image Processor ("RIP") filters that convert PostScript or image files into bitmaps that can be sent to a raster printer.

Backends

Backends perform the most important task of all − they send the filtered print data to the printer.

CUPS provides backends for printing over parallel, serial, and USB ports, and over the network via the IPP, JetDirect (AppSocket), and Line Printer Daemon ("LPD") protocols. Additional backends are available in network service packages such as the SMB backend included with the popular SAMBA software.

Backends are also used to determine the available devices. On startup each backend is asked for a list of devices it supports, and any information that is available. This allows the parallel backend to tell CUPS that an EPSON Stylus Color 600 printer is attached to parallel port 1, for example.

Printer Drivers

Printer drivers in CUPS consist of one of more filters specific to a printer. CUPS includes sample printer drivers for Hewlett−Packard LaserJet and DeskJet printers and EPSON 9−pin, 24−pin, Stylus Color, and Stylus Photo printers. While these drivers do not generate optimal output for the different printer models, they do provide basic printing and demonstrate how you can write your own printer drivers and incorporate them into CUPS.

Networking

Printers and classes on the local system are automatically shared with other systems on the network. This allows you to setup one system to print to a printer and use this system as a printer server or spool host for all of the others. Users may then select a local printer by name or a remote printer using "name@server".

CUPS also provides *implicit classes*, which are collections of printers and/or classes with the same name. This allows you to setup multiple servers pointing to the same physical network printer, for example, so that you aren't relying on a single system for printing. Because this also works with printer classes, you can setup multiple servers and printers and never worry about a single point of failure unless all of the printers and servers go down!

2 – Building and Installing CUPS

This chapter shows how to build and install the Common UNIX Printing System. If you are installing a binary distribution from the CUPS web site, proceed to the section titled, Installing a Binary Distribution.

Installing a Source Distribution

This section describes how to compile and install CUPS on your system from the source code.

Requirements

You'll need ANSI–compliant C and C++ compilers to build CUPS on your system. As its name implies, CUPS is designed to run on the UNIX operating system, however the CUPS interface library and most of the filters and backends supplied with CUPS should also compile and run under Microsoft Windows.

For the image file filters and PostScript RIP, you'll need the JPEG, PNG, TIFF, and ZLIB libraries. CUPS will build without these, but with significantly reduced functionality. Easy Software Products maintains a mirror of the current versions of these libraries at:

        ftp://ftp.easysw.com/pub/libraries

If you make changes to the man pages you'll need GNU groff or another nroff–like package. GNU groff is available from:

        ftp://ftp.gnu.org/pub/groff

The documentation is formatted using the HTMLDOC software. If you need to make changes you can get the HTMLDOC software from:

```
http://www.easysw.com/htmldoc
```

Finally, you'll need a `make` program that understands the `include` directive – FreeBSD, NetBSD, and OpenBSD developers should use the `gmake` program.

Compiling CUPS

CUPS uses GNU autoconf to configure the makefiles and source code for your system. Type the following command to configure CUPS for your system:

```
./configure ENTER
```

The default installation will put the CUPS software in the */etc*, */usr*, and */var* directories on your system, which will overwrite any existing printing commands on your system. Use the `--prefix` option to install the CUPS software in another location:

```
./configure --prefix=/some/directory ENTER
```

If the PNG, JPEG, TIFF, and ZLIB libraries are not installed in a system default location (typically */usr/include* and */usr/lib*) you'll need to set the `CFLAGS`, `CXXFLAGS`, and `LDFLAGS` environment variables prior to running configure:

```
setenv CFLAGS "-I/some/directory" ENTER
setenv CXXFLAGS "-I/some/directory" ENTER
setenv LDFLAGS "-L/some/directory" ENTER
setenv DSOFLAGS "-L/some/directory" ENTER
./configure ... ENTER
```

or:

```
CFLAGS="-I/some/directory"; export CFLAGS ENTER
CXXFLAGS="-I/some/directory"; export CXXFLAGS ENTER
LDFLAGS="-L/some/directory"; export LDFLAGS ENTER
DSOFLAGS="-L/some/directory"; export DSOFLAGS ENTER
./configure ... ENTER
```

To enable support for encryption, you'll also want to add the "--enable-ssl" option:

```
./configure --enable-ssl
```

SSL and TLS support require the OpenSSL library, available at:

```
http://www.openssl.org
```

If the OpenSSL headers and libraries are not installed in the standard directories, use the `--with-openssl-includes` and `--with-openssl-libs` options:

```
./configure --enable-ssl \
    --with-openssl-includes=/foo/bar/include \
    --with-openssl-libs=/foo/bar/lib
```

Once you have configured things, just type:

> **make ENTER**

to build the software.

Installing the Software

Use the "install" target to install the software:

> **make install ENTER**

> **WARNING:**
>
> Installing CUPS will overwrite your existing printing system. If you experience difficulties with the CUPS software and need to go back to your old printing system, you will need to reinstall the old printing system from your operating system CDs.

Running the Software

Once you have installed the software you can start the CUPS server by typing:

> **/usr/sbin/cupsd ENTER**

Installing a Binary Distribution

CUPS comes in a variety of binary distribution formats. Easy Software Products provides binaries in TAR format with installation and removal scripts ("portable" distributions), and in RPM and DPKG formats for Red Hat and Debian−based distributions. Portable distributions are available for all platforms, while the RPM and DPKG distributions are only available for Linux.

> **WARNING:**
>
> Installing CUPS will overwrite your existing printing system. If you experience difficulties with the CUPS software and need to go back to your old printing system, you will need to remove the CUPS software with the provided script and/or reinstall the old printing system from your operating system CDs.

Installing a Portable Distribution

To install the CUPS software from a portable distribution you will need to be logged in as root; doing an su is good enough. Once you are the root user, run the installation script with:

> **./cups.install ENTER**

After asking you a few yes/no questions the CUPS software will be installed and the scheduler will be started automatically.

Installing an RPM Distribution

To install the CUPS software from an RPM distribution you will need to be logged in as root; doing an su is good enough. Once you are the root user, run RPM with:

```
rpm -e lpr
rpm -i cups-1.1-linux-M.m.n-intel.rpm ENTER
```

After a short delay the CUPS software will be installed and the scheduler will be started automatically.

Installing an Debian Distribution

To install the CUPS software from a Debian distribution you will need to be logged in as root; doing an su is good enough. Once you are the root user, run dpkg with:

```
dpkg -i cups-1.1-linux-M.m.n-intel.deb ENTER
```

After a short delay the CUPS software will be installed and the scheduler will be started automatically.

3 – Managing Printers

This chapter describes how to add your first printer and how to manage your printers.

The Basics

Each printer queue has a name associated with it; the printer name must start with any printable character except " ", "/", and "@". It can contain up to 127 letters, numbers, and the underscore (_). Case is not significant, e.g. "PRINTER", "Printer", and "printer" are considered to be the same name.

Printer queues also have a device associated with them. The device can be a parallel port, a network interface, and so forth. Devices within CUPS use Uniform Resource Identifiers ("URIs") which are a more general form of Uniform Resource Locators ("URLs") that are used in your web browser. For example, the first parallel port in Linux usually uses a device URI of `parallel:/dev/lp1`.

You can see a complete list of supported devices by running the `lpinfo(8)` command:

```
lpinfo -v ENTER
network socket
network http
network ipp
network lpd
direct parallel:/dev/lp1
serial serial:/dev/ttyS1?baud=115200
serial serial:/dev/ttyS2?baud=115200
direct usb:/dev/usb/lp0
network smb
```

The `-v` option specifies that you want a list of available devices. The first word in each line is the type of device (direct, file, network, or serial) and is followed by the device URI or method name for that device. File devices have device URIs of the form `file:/directory/filename` while network devices use the more familiar `method://server` or `method://server/path` format.

Finally, printer queues usually have a PostScript Printer Description ("PPD") file associated with them. PPD files describe the capabilities of each printer, the page sizes supported, etc., and are used for PostScript and non−PostScript printers. CUPS includes PPD files for HP LaserJet, HP DeskJet, EPSON 9−pin, EPSON 24−pin, and EPSON Stylus printers.

Adding Your First Printer

CUPS provides two methods for adding printers: a command−line program called `lpadmin(8)` and a Web interface. The `lpadmin` command allows you to perform most printer administration tasks from the command−line and is located in *∕usr∕sbin*. The Web interface is located at:

> http://localhost:631/admin

and steps you through printer configuration. If you don't like command−line interfaces, try the Web interface instead.

Adding Your First Printer from the Command−Line

Run the `lpadmin` command with the `-p` option to add a printer to CUPS:

> **/usr/sbin/lpadmin -p *printer* -E -v *device* -m *ppd* ENTER**

For a HP DeskJet printer connected to the parallel port this would look like:

> **/usr/sbin/lpadmin -p DeskJet -E -v parallel:/dev/lp1 -m deskjet.ppd ENTER**

Similarly, a HP LaserJet printer using a JetDirect network interface at IP address 11.22.33.44 would be added with the command:

> **/usr/sbin/lpadmin -p LaserJet -E -v socket://11.22.33.44 -m laserjet.ppd ENTER**

As you can see, `deskjet.ppd` and `laserjet.ppd` are the PPD files for the HP DeskJet and HP LaserJet drivers included with CUPS. You'll find a complete list of PPD files and the printers they will work with in Appendix C, "Printer Drivers".

For a dot matrix printer connected to the serial port this would might look like:

```
/usr/sbin/lpadmin -p DotMatrix -E -v serial:/dev/ttyS0?baud=9600+size=8+parity=none+flow=s
```

Here you specify the serial port (e.g. S0,S1, d0, d1), baud rate (e.g. 9600, 19200, 38400, 115200, etc.), number of bits, parity, and flow control. If you do not need flow control, delete the "+flow=soft" portion.

Adding Your First Printer from the Web

The CUPS web server provides a user–friendly "wizard" interface for adding your printers. Rather than figuring out which device URI and PPD file to use, you can instead click on the appropriate listings and fill in some simple information. Enter the following URL in your web browser to begin:

http://localhost:631/admin

Click on the *Add Printer* button to add a printer.

Managing Printers from the Command–Line

The lpadmin command enables you to perform most printer administration tasks from the command–line. You'll find lpadmin in the */usr/sbin* directory.

Adding and Modifying Printers

Run the lpadmin command with the −p option to add or modify a printer:

```
/usr/sbin/lpadmin -p printer options ENTER
```

The *options* arguments can be any of the following:

−c *class*
> Adds the named printer to printer class *class*. If the class does not exist then it is created.

−i *interface*
> Copies the named *interface* script to the printer. Interface scripts are used by System V printer drivers. Since all filtering is disabled when using an interface script, scripts generally should not be used unless there is no other driver for a printer.

−m *model*
> Specifies a standard printer driver which is usually a PPD file. PPD's are usually stored under the */usr/share/cups/model/* directory structure. A list of all available models can be displayed using the lpinfo command with the −m option. A list of printer drivers included with CUPS can be found in Appendix C, "Printer Drivers".

−r *class*
> Removes the named printer from printer class *class*. If the resulting class becomes empty then it is removed.

−v *device–uri*
> Sets the device for communicating with the printer. If a job is currently printing on the named printer then the job will be restarted and sent to the new device.

−D *info*
> Provides a textual description of the printer, e.g. "John's Personal Printer".

−E

Enables the printer and accepts job. This option is equivalent to running the `enable(1)` and `accept(8)` commands on the printer.

−L location
Provides a textual location for the printer, e.g. "Computer Lab 5".

−P ppd−file
Specifies a local PPD file for the printer driver.

Deleting Printers

Run the `lpadmin` command with the `-x` option to delete a printer:

```
/usr/sbin/lpadmin -x printer ENTER
```

Setting the Default Printer

Run the `lpadmin` command with the `-d` option to set a default printer:

```
/usr/sbin/lpadmin -d printer ENTER
```

The default printer can be overridden by the user using the `lpoptions(1)` command.

Starting and Stopping Printers

The `enable` and `disable` commands start and stop printer queues, respectively:

```
/usr/bin/enable printer ENTER
/usr/bin/disable printer ENTER
```

Printers that are disabled may still accept jobs for printing, but won't actually print any files until they are restarted. This is useful if the printer malfunctions and you need time to correct the problem. Any queued jobs are printed after the printer is enabled (started).

Accepting and Rejecting Print Jobs

The `accept` and `reject` commands accept and reject print jobs for the named printer, respectively:

```
/usr/sbin/accept printer ENTER
/usr/sbin/reject printer ENTER
```

As noted above, a printer can be stopped but accepting new print jobs. A printer can also be rejecting new print jobs while it finishes those that have been queued. This is useful for when you must perform maintenance on the printer and will not have it available to users for a long period of time.

Setting Quotas on a Printer

CUPS supports page and size−based quotas for each printer. The quotas are tracked individually for each user, but a single set of limits applies to all users for a partiuclar printer. For example, you can limit every user to 5 pages per day on an expensive printer, but you cannot limit every user except Johnny.

The *job−k−limit*, *job−page−limit*, and *job−quota−peiod* options determine whether and how quotas are

enforced for a printer. The *job−quota−period* option determines the time interval for quota tracking. The interval is expressed in seconds, so a day is 86,400, a week is 604,800 and a month is 2,592,000 seconds. The *job−k−limit* option specifies the job size limit in killobytes. The *job−page−limit* option specifies the number of pages limit.

For quotas to be enforced, the period and at least one of the limits must be set to a non−zero value. The following options will enable quotas:

```
/usr/sbin/lpadmin -p printer -o job-quota-period=604800 -o job-k-limit=1024 ENTER
/usr/sbin/lpadmin -p printer -o job-quota-period=604800 -o job-page-limit=100 ENTER
```

Or, you can combine all three options on the same line.

Restricting User Access to a Printer

The `-u` option of the `lpadmin` command controls which users can print to a printer. The default configuration allows all users to print to a printer:

```
/usr/sbin/lpadmin -p printer -u allow:all ENTER
```

CUPS supports allow and deny lists so that you can specify a list of users who are allowed to print or not allowed to print. Along with your list of users, you can specify whether they are allowed or not allowed to use the printer:

```
/usr/sbin/lpadmin -p printer -u allow:peter,paul,mary ENTER
```

This command allows peter, paul, and mary to print to the named printer, but all other users cannot print. The command:

```
/usr/sbin/lpadmin -p printer -u deny:peter,paul,mary ENTER
```

has the opposite effect. All users except peter, paul, and mary will be able to print to the named printer.

> **NOTE:**
>
> The *allow* and *deny* options are not cummulative. That is, you must provide the complete list of users to allow or deny each time.
>
> Also, CUPS only maintains one list of users – the list can allow or deny users from printing. If you specify an allow list and then specify a deny list, the deny list will replace the allow list – only one list is active at any time.

Managing Printers from the Web

The Web interface is located at:

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

From there you can perform all printer management tasks with a few simple mouse clicks.

Restricting User Access to a Printer

4 – Printer Classes

This chapter describes what printer classes are and how to manage them.

The Basics

CUPS provides collections of printers called *printer classes*. Jobs sent to a class are forwarded to the first available printer in the class. Classes can themselves be members of other classes, so it is possible for you to define very large, distributed printer classes for high–availability printing.

CUPS also supports *implicit classes*. Implicit classes work just like printer classes, but they are created automatically based upon the available printers and classes on the network. This allows you to setup multiple print servers with identical printer configurations and have the client machines send their print jobs to the first available server. If one or more servers go down, the jobs are automatically redirected to the servers that are running, providing fail–safe printing.

Managing Printer Classes from the Command–Line

Run the `lpadmin` command with the `-p` and `-c` options to add a printer to a class:

```
/usr/sbin/lpadmin -p printer -c class ENTER
```

The *class* is created automatically if it doesn't exist. To remove a printer from a class use the `-r` option:

```
/usr/sbin/lpadmin -p printer -r class ENTER
```

To remove the entire class just use the −x option:

    /usr/sbin/lpadmin −x *class* **ENTER**

Managing Printer Classes from the Web Interface

The Web interface is located at:

    http://localhost:631/admin

The *Add Class* and *Modify Class* interfaces provide a list of available printers; click on the printers of interest to add them to the class.

Implicit Classes

A noted earlier, implicit classes are created automatically from the available network printers and classes. To disable this functionality, set the ImplicitClasses directive to Off in the cupsd.conf file. You will find more information on doing this in Chapter 6, "Printing System Management".

5 – Client Setup

This chapter discusses several ways to configure CUPS clients for printing.

The Basics

A client is any machine that sends print jobs to another machine for final printing. Clients can also be servers if they communicate directly with any printers of their own.

CUPS supports several methods of configuring client machines:

- Manual configuration of print queues.
- Specifying a single server for printing.
- Automatic configuration of print queues.
- Specifying multiple servers for printing.
- Relaying printers to other clients.

Manual Configuration of Print Queues

The most tedious method of configuring client machines is to configure each remote queue by hand using the `lpadmin` command:

        lpadmin -p *printer* -E -v ipp://*server*/printers/*printer* ENTER

The `printer` name is the name of the printer on the server machine. The `server` name is the hostname or

IP address of the server machine. Repeat the `lpadmin` command for each remote printer you wish to use.

> **NOTE:**
>
> Manual configuration of print queues is not recommended for large numbers of client machines because of the administration nightmare it creates. For busy networks, consider subnetting groups of clients and polling and relaying printer information instead.

Specifying a Single Server for Printing

CUPS can be configured to run without a local spooler and send all jobs to a single server. However, if that server goes down then all printing will be disabled. Use this configuration only as absolutely needed.

The default server is normally "localhost". To override the default server create a file named */etc/cups/client.conf* and add a line reading:

```
ServerName server
```

to the file. The *server* name can be the hostname or IP address of the default server.

The default server can also be customized on a per−user basis. To set a user−specific server create a file named *~/.cupsrc* and add a line reading:

```
ServerName server
```

to the file. The *server* name can be the hostname or IP address of the default server.

Automatic Configuration of Print Queues

CUPS supports automatic client configuration of printers on the same subnet. To configure printers on the same subnet, *do nothing*. Each client should see the available printers within 30 seconds automatically. The printer and class lists are updated automatically as printers and servers are added or removed.

If you want to see printers on other subnets as well, use the `BrowsePoll` directive as described next.

> **NOTE:**
>
> The `BrowseAddress` directive enables broadcast traffic from your server. The default configuration braodcasts printer information every 30 seconds. Although this printer information does not use much bandwidth, typically about 80 bytes per printer, it can add up with large numbers of servers and printers.
>
> Use the `BrowseInterval` and `BrowseTimeout` directives to tune the amount of data that is added to your network load. In addition, subnets can be used to minimize the amount of traffic that is carried by the "backbone" of your large network.

Specifying Multiple Servers for Printing

If you have CUPS servers on different subnets, then you should configure CUPS to poll those servers. Polling provides the benefits of automatic configuration without significant configuration on the clients, and multiple clients on the same subnet can share the same configuration information.

Polling is enabled by specifying one or more `BrowsePoll` directives in the */etc/cups/cupsd.conf* file. For information on making these changes, see Chapter 6, "Printing System Management".

Multiple `BrowsePoll` lines can be used to poll multiple CUPS servers. To limit the amount of polling you do from client machines, you can have only one of the clients do the polling and relay that information to the others on the same subnet (described next).

Relaying Printers to Other Clients

When you have clients and servers spread across multiple subnets, the polling method is inefficient. CUPS provides a `BrowseRelay` directive that enables a single client to relay (broadcast) the polled printer information to the local subnet.

For example, Server A and Server B are on subnet 1 and subnet 2, while the clients are on subnet 3. To provide printers to all of the clients in subnet 3, client C will be configured with the following directives in */etc/cups/cupsd.conf*:

```
# Poll the two servers

BrowsePoll ServerA ENTER
BrowsePoll ServerB ENTER


# Relay the printers to the local subnet

BrowseRelay 127.0.0.1 192.168.3.255 ENTER
```

The `BrowseRelay` line specifies a source address and mask. Any browse packets coming from a matching address wil be sent to the given broadcast address. In this case, we want the packets from the local machine (127.0.0.1) relayed to the other clients.

As printers are found using polling, they are relayed from client C to the rest of the clients through a broadcast on subnet 3. The rest of the clients can use the standard *cupsd.conf* configuration.

The `BrowseRelay` directive can also be used to relay browsing packets from one network interface to another. For example, if client C in the previous example had network interfaces attaches to both subnet 1 and subnet 2, it could use the `BrowseRelay` directive exclusively:

```
# Relay the printers from subnet 1 and 2 to subnet 3

BrowseRelay 192.168.1 192.168.3.255 ENTER
BrowseRelay 192.168.2 192.168.3.255 ENTER
```

Load Balancing and Failsafe Operation

When using server polling or broadcasting, CUPS clients can automatically merge identical printers on multiple servers into a single *implicit class* queue. Clients assume that printers with the same name on multiple servers are in fact the same printer or type of printer being served by multiple machines.

If you have two printers, LaserJet@ServerA and LaserJet@ServerB, a third implicit class called *LaserJet* will be created automatically on the client that refers to both printers. If the client also has a local printer with the name LaserJet then an implicit class named *AnyLaserJet* will be created instead.

The client will alternate between servers and automatically stop sending jobs to a server if it goes down, providing a load–balancing effect and fail–safe operation with automatic switchover.

> **NOTE:**
>
> Note that implicit classes (`ImplicitClasses`) are enabled by default.

6 – Printing System Management

This chapter shows how you can configure the CUPS server.

The Basics

Several text files are used to configure CUPS. All of the server configuration files are located in the *etc/cups* directory:

classes.conf
> This file contains information on each printer class. Normally you manipulate this file using the `lpadmin` command or the Web interface.

client.conf
> This file provides the default server name for client machines. See for more information.

cupsd.conf
> This file controls how the CUPS server (*/usr/sbin/cupsd*) operates and is normally edited by hand.

mime.convs

This file contains a list of standard file conversion filters and their costs. You normally do not edit this file.

mime.types

This file contains a list of standard file formats and how to recognize them. You normally do not edit this file.

printers.conf

This file contains information on each printer. Normally you manipulate this file using the `lpadmin` command or the Web Interface.

Restarting the CUPS Server

Once you have made a change to a configuration file you need to restart the CUPS server by sending it a `HUP` signal or using the supplied initialization script. The CUPS distributions install the script in the *init.d* directory with the name *cups*. The location varies based upon the operating system:

```
/etc/software/init.d/cups restart ENTER
/etc/rc.d/init.d/cups restart ENTER
/etc/init.d/cups restart ENTER
/sbin/init.d/cups restart ENTER
```

Changing the Server Configuration

The */etc/cups/cupsd.conf* file contains configuration *directives* that control how the server functions. Each directive is listed on a line by itself followed by its value. Comments are introduced using the number sign ("#") character at the beginning of a line. Since the server configuration file consists of plain text, you can use your favorite text editor to make changes to it.

Server Directives

The *cupsd.conf* file contains many directives that determine how the server operates:

- AccessLog
- Allow
- AuthClass
- AuthGroupName
- AuthType
- AutoPurgeJobs
- BrowseAddress
- BrowseAllow
- BrowseDeny
- BrowseInterval
- BrowseOrder
- BrowsePoll
- BrowsePort
- BrowseProtocols
- BrowseRelay
- BrowseShortNames
- BrowseTimeout
- Browsing
- Classification
- ClassifyOverride
- ConfigFilePerm
- DataDir
- DefaultCharset
- DefaultLanguage
- Deny
- DocumentRoot
- Encryption

- ErrorLog
- FileDevice
- FilterLimit
- FilterNice
- FontPath
- Group
- HideImplicitMembers
- HostNameLookups
- ImplicitClasses
- ImplicitAnyClasses
- Include
- KeepAliveTimeout
- KeepAlive
- Limit
- LimitExcept
- LimitRequestBody
- Listen
- Location
- LogFilePerm
- LogLevel
- MaxClients
- MaxClientsPerHost
- MaxCopies
- MaxJobs
- MaxJobsPerPrinter
- MaxJobsPerUser
- MaxLogSize
- MaxRequestSize
- Order
- PageLog

- Port
- PreserveJobFiles
- PreserveJobHistory
- Printcap
- PrintcapFormat
- PrintcapGUI
- RemoteRoot
- RequestRoot
- Require
- RIPCache
- RootCertDuration
- RunAsUser
- Satisfy
- ServerAdmin
- ServerBin
- ServerCertificate
- ServerKey
- ServerName
- ServerRoot
- SSLListen
- SSLPort
- SystemGroup
- TempDir
- Timeout
- User

AccessLog

Examples

```
AccessLog /var/log/cups/access_log
AccessLog /var/log/cups/access_log-%s
AccessLog syslog
```

Description

The AccessLog directive sets the name of the access log file. If the filename is not absolute then it is assumed to be relative to the ServerRoot directory. The access log file is stored in "common log format" and can be used by any web access reporting tool to generate a report on CUPS server activity.

The server name can be included in the filename by using %s in the name.

The special name "syslog" can be used to send the access information to the system log instead of a plain file.

The default access log file is */var/log/cups/access_log*.

Allow

Examples

```
Allow from All
Allow from None
Allow from *.domain.com
Allow from .domain.com
Allow from host.domain.com
Allow from nnn.*
Allow from nnn.nnn.*
Allow from nnn.nnn.nnn.*
Allow from nnn.nnn.nnn.nnn
Allow from nnn.nnn.nnn.nnn/mm
Allow from nnn.nnn.nnn.nnn/mmm.mmm.mmm.mmm
Allow from @LOCAL
Allow from @IF(name)
```

Description

The `Allow` directive specifies a hostname, IP address, or network that is allowed access to the server. `Allow` directives are cummulative, so multiple `Allow` directives can be used to allow access for multiple hosts or networks. The `/mm` notation specifies a CIDR netmask:

mm	netmask	mm	netmask
0	0.0.0.0	8	255.0.0.0
1	128.0.0.0	16	255.255.0.0
2	192.0.0.0	24	255.255.255.0
...	...	32	255.255.255.255

The `@LOCAL` name will allow access from all local network interfaces, but not remote point−to−point interfaces. The `@IF(name)` name will allow access from the named interface.

The `Allow` directive must appear inside a <u>Location</u> directive.

AuthClass

Examples

```
AuthClass Anonymous
AuthClass User
AuthClass System
AuthClass Group
```

Description

The `AuthClass` directive defines what level of authentication is required:

- `Anonymous` – No authentication should be performed (default.)
- `User` – A valid username and password is required.
- `System` – A valid username and password is required, and the username must belong to the "sys" group; this can be changed using the <u>SystemGroup</u> directive.
- `Group` – A valid username and password is required, and the username must belong to the group named by the `AuthGroupName` directive.

The `AuthClass` directive must appear inside a <u>Location</u> directive.

AuthGroupName

Examples

```
AuthGroupName mygroup
AuthGroupName lp
```

Description

The `AuthGroupName` directive sets the group to use for `Group` authentication.

The `AuthGroupName` directive must appear inside a <u>Location</u> directive.

AuthType

Examples

```
AuthType None
AuthType Basic
AuthType Digest
AuthType BasicDigest
```

Description

The `AuthType` directive defines the type of authentication to perform:

- `None` – No authentication should be performed (default.)
- `Basic` – Basic authentication should be performed using the UNIX password and group files.
- `Digest` – Digest authentication should be performed using the *etc/cups/passwd.md5* file.

AuthClass

- `BasicDigest` – Basic authentication should be performed using the *etc/cups/passwd.md5* file.

When using `Basic`, `Digest`, or `BasicDigest` authentication, clients connecting through the `localhost` interface can also authenticate using certificates.

The `AuthType` directive must appear inside a <u>Location</u> directive.

AutoPurgeJobs

Examples

```
AutoPurgeJobs Yes
AutoPurgeJobs No
```

Description

The `AutoPurgeJobs` directive specifies whether or not to purge completed jobs once they are no longer required for quotas. This option has no effect if quotas are not enabled. The default setting is `No`.

BrowseAddress

Examples

```
BrowseAddress 255.255.255.255:631
BrowseAddress 192.0.2.255:631
BrowseAddress host.domain.com:631
BrowseAddress @LOCAL
BrowseAddress @IF(name)
```

Description

The `BrowseAddress` directive specifies an address to send browsing information to. Multiple `BrowseAddress` directives can be specified to send browsing information to different networks or systems.

The `@LOCAL` name will broadcast printer information to all local interfaces. The `@IF(name)` name will broadcast to the named interface.

No browse addresses are set by default.

> **NOTE:**
>
> If you are using HP−UX 10.20 and a subnet that is not 24, 16, or 8 bits, printer browsing (and in fact all broadcast reception) will not work. This problem appears to be fixed in HP−UX 11.0.

BrowseAllow

Examples

```
BrowseAllow from all
BrowseAllow from none
BrowseAllow from 192.0.2
BrowseAllow from 192.0.2.0/24
BrowseAllow from 192.0.2.0/255.255.255.0
BrowseAllow from *.domain.com
BrowseAllow from @LOCAL
BrowseAllow from @IF(name)
```

Description

The `BrowseAllow` directive specifies a system or network to accept browse packets from. The default is to accept browse packets from all hosts.

Host and domain name matching require that you enable the <u>HostNameLookups</u> directive.

IP address matching supports exact matches, partial addresses that match networks using netmasks of 255.0.0.0, 255.255.0.0, and 255.255.255.0, or network addresses using the specified netmask or bit count.

The `@LOCAL` name will allow browse data from all local network interfaces, but not remote point−to−point interfaces. The `@IF(name)` name will allow browse data from the named interface.

BrowseDeny

Examples

```
BrowseDeny from all
BrowseDeny from none
BrowseDeny from 192.0.2
BrowseDeny from 192.0.2.0/24
BrowseDeny from 192.0.2.0/255.255.255.0
BrowseDeny from *.domain.com
BrowseDeny from @LOCAL
BrowseDeny from @IF(name)
```

Description

The `BrowseDeny` directive specifies a system or network to reject browse packets from. The default is to deny browse packets from no hosts.

Host and domain name matching require that you enable the <u>HostNameLookups</u> directive.

IP address matching supports exact matches, partial addresses that match networks using netmasks of 255.0.0.0, 255.255.0.0, and 255.255.255.0, or network addresses using the specified netmask or bit count.

The `@LOCAL` name will block browse data from all local network interfaces, but not remote point−to−point interfaces. The `@IF(name)` name will block browse data from the named interface.

BrowseOrder

Examples

```
BrowseOrder allow,deny
BrowseOrder deny,allow
```

Description

The `BrowseOrder` directive specifies the order of allow/deny processing. The default order is `deny,allow`:

- `allow,deny` – Browse packets are accepted unless specifically denied.
- `deny,allow` – Browse packets are rejected unless specifically allowed.

BrowseInterval

Examples

```
BrowseInterval 0
BrowseInterval 30
```

Description

The `BrowseInterval` directive specifies the maximum amount of time between browsing updates. Specifying a value of 0 seconds disables outgoing browse updates but allows a server to receive printer information from other hosts.

The `BrowseInterval` value should always be less than the `BrowseTimeout` value. Otherwise printers and classes will disappear from client systems between updates.

BrowsePoll

Examples

```
BrowsePoll 192.0.2.2:631
BrowsePoll host.domain.com:631
```

Description

The `BrowsePoll` directive polls a server for available printers once every `BrowseInterval` seconds. Multiple `BrowsePoll` directives can be specified to poll multiple servers.

If `BrowseInterval` is set to 0 then the server is polled once every 30 seconds.

BrowsePort

Examples

```
BrowsePort 631
BrowsePort 9999
```

Description

The `BrowsePort` directive specifies the UDP port number used for browse packets. The default port number is 631.

> **NOTE:**
>
> You must set the `BrowsePort` to the same value on all of the systems that you want to see.

BrowseProtocols

Examples

```
BrowseProtocols CUPS
BrowseProtocols SLP
BrowseProtocols CUPS SLP
BrowseProtocols all
```

Description

The `BrowseProtocols` directive specifies the protocols to use when collecting and distributing shared printers on the local network. The default protocol is `CUPS`, which is a broadcast–based protocol.

> **NOTE:**
>
> When using the `SLP` protocol, you must have at least one Directory Agent (DA) server on your network. Otherwise the CUPS scheduler (`cupsd`) will not respond to client requests for several seconds while polling the network.

BrowseRelay

Examples

```
BrowseRelay 193.0.2.1 192.0.2.255
BrowseRelay 193.0.2.0/255.255.255.0 192.0.2.255
BrowseRelay 193.0.2.0/24 192.0.2.255
BrowseRelay *.domain.com 192.0.2.255
BrowseRelay host.domain.com 192.0.2.255
```

Description

The `BrowseRelay` directive specifies source and destination addresses for relaying browsing information from one host or network to another. Multiple `BrowseRelay` directives can be specified as needed.

`BrowseRelay` is typically used on systems that bridge multiple subnets using one or more network interfaces. It can also be used to relay printer information from polled servers with the line:

```
BrowseRelay 127.0.0.1 255.255.255.255
```

This effectively provides access to printers on a WAN for all clients on the LAN(s).

BrowseShortNames

Examples

```
BrowseShortNames Yes
BrowseShortNames No
```

Description

The `BrowseShortNames` directive specifies whether or not short names are used for remote printers when possible. Short names are just the remote printer name, without the server ("printer"). If more than one remote printer is detected with the same name, the printers will have long names ("printer@server1", "printer@server2".)

The default value for this option is `Yes`.

BrowseTimeout

Examples

```
BrowseTimeout 300
BrowseTimeout 60
```

Description

The `BrowseTimeout` directive sets the timeout for printer or class information that is received in browse packets. Once a printer or class times out it is removed from the list of available destinations.

The `BrowseTimeout` value should always be greater than the <u>BrowseInterval</u> value. Otherwise printers and classes will disappear from client systems between updates.

Browsing

Examples

```
Browsing On
Browsing Off
```

Description

The `Browsing` directive controls whether or not network printer browsing is enabled. The default setting is `On`.

> **NOTE:**
>
> If you are using HP–UX 10.20 and a subnet that is not 24, 16, or 8 bits, printer browsing (and in fact all broadcast reception) will not work. This problem appears to be fixed in HP–UX 11.0.

Classification

Examples

```
Classification
Classification classified
Classification confidential
Classification secret
Classification topsecret
Classification unclassified
```

Description

The `Classification` directive sets the classification level on the server. When this option is set, at least one of the banner pages is forced to the classification level, and the classification is placed on each page of output. The default is no classification level.

ClassifyOverride

Examples

```
ClassifyOverride Yes
ClassifyOverride No
```

Description

The `ClassifyOverride` directive specifies whether users can override the default classification level on the server. When the server classification is set, users can change the classification using the `job-sheets` option and can choose to only print one security banner before or after the job. If the `job-sheets` option is set to `none` then the server default classification is used.

The default is to not allow classification overrides.

ConfigFilePerm

Examples

```
ConfigFilePerm 0644
ConfigFilePerm 0600
```

Description

The `ConfigFilePerm` directive specifies the permissions to use when writing configuration files. The default is 0600.

DataDir

Examples

```
DataDir /usr/share/cups
```

Description

The `DataDir` directive sets the directory to use for data files.

DefaultCharset

Examples

```
DefaultCharset utf-8
DefaultCharset iso-8859-1
DefaultCharset windows-1251
```

Description

The `DefaultCharset` directive sets the default character set to use for client connections. The default character set is `utf-8` but is overridden by the character set for the language specified by the client or the `DefaultLanguage` directive.

DefaultLanguage

Examples

```
DefaultLanguage de
DefaultLanguage en
DefaultLanguage es
DefaultLanguage fr
DefaultLanguage it
```

Description

The `DefaultLanguage` directive specifies the default language to use for client connections. Setting the default language also sets the default character set if a language localization file exists for it. The default language is "en" for English.

Deny

Examples

```
Deny from All
Deny from None
Deny from *.domain.com
Deny from .domain.com
Deny from host.domain.com
Deny from nnn.*
Deny from nnn.nnn.*
Deny from nnn.nnn.nnn.*
Deny from nnn.nnn.nnn.nnn
Deny from nnn.nnn.nnn.nnn/mm
Deny from nnn.nnn.nnn.nnn/mmm.mmm.mmm.mmm
Deny from @LOCAL
Deny from @IF(name)
```

Description

The `Deny` directive specifies a hostname, IP address, or network that is allowed access to the server. `Deny` directives are cummulative, so multiple `Deny` directives can be used to allow access for multiple hosts or networks. The `/mm` notation specifies a CIDR netmask:

mm	netmask	mm	netmask
0	0.0.0.0	8	255.0.0.0
1	128.0.0.0	16	255.255.0.0
2	192.0.0.0	24	255.255.255.0
...	...	32	255.255.255.255

The `@LOCAL` name will deny access from all local network interfaces, but not remote point−to−point interfaces. The `@IF(name)` name will deny access from the named interface.

The `Deny` directive must appear inside a <u>Location</u> directive.

DocumentRoot

Examples

```
DocumentRoot /usr/share/doc/cups
DocumentRoot /foo/bar/doc/cups
```

Description

The `DocumentRoot` directive specifies the location of web content for the HTTP server in CUPS. If an absolute path is not specified then it is assumed to be relative to the <u>ServerRoot</u> directory. The default directory is */usr/share/doc/cups.*

Documents are first looked up in a sub−directory for the primary language requested by the client (e.g. */usr/share/doc/cups/fr/...*) and then directly under the `DocumentRoot` directory (e.g. */usr/share/doc/cups/...*), so it is possible to localize the web content by providing subdirectories for each language needed.

Encryption

Examples

```
Encryption Never
Encryption IfRequested
Encryption Required
Encryption Always
```

Description

The `Encryption` directive must appear instead a <u>Location</u> section and specifies the encryption settings for that location. The default setting is `IfRequested` for all locations.

ErrorLog

Examples

```
ErrorLog /var/log/cups/error_log
ErrorLog /var/log/cups/error_log-%s
ErrorLog syslog
```

Description

The `ErrorLog` directive sets the name of the error log file. If the filename is not absolute then it is assumed to be relative to the <u>ServerRoot</u> directory. The default error log file is */var/log/cups/error_log*.

The server name can be included in the filename by using `%s` in the name.

The special name "syslog" can be used to send the error information to the system log instead of a plain file.

FileDevice

Examples

```
FileDevice Yes
FileDevice No
```

Description

The `FileDevice` directive determines whether the scheduler allows new printers to be added using device URIs of the form `file:/filename`. File devices are most often used to test new printer drivers and do no support raw file printing.

The default setting is `No`.

> **Note:**
>
> File devices are managed by the scheduler. Since the scheduler normally runs as the root user, file devices can be used to overwrite system files and potentially gain

unauthorized access to the system. If you must create printers using file devices, we recommend that you set the `FileDevice` directive to `Yes` for only as long as you need to add the printers to the system, and then reset the directive to `No`.

FilterLimit

Examples

```
FilterLimit 0
FilterLimit 200
FilterLimit 1000
```

Description

The `FilterLimit` directive sets the maximum cost of all running job filters. It can be used to limit the number of filter programs that are run on a server to minimize disk, memory, and CPU resource problems. A limit of 0 disables filter limiting.

An average print to a non−PostScript printer needs a filter limit of about 200. A PostScript printer needs about half that (100). Setting the limit below these thresholds will effectively limit the scheduler to printing a single job at any time.

The default limit is 0.

FilterNice

Examples

```
FilterNice 0
FilterNice 39
FilterNice −10
```

Description

The `FilterNice` directive sets the scheduling priority of job filters. Values larger than 0 give filters a lower priority while values smaller than 0 give filters a higher priority. The `FilterNice` value does not affect the priority of job backends.

The default priority is 0.

FontPath

Examples

```
FontPath /foo/bar/fonts
FontPath /usr/share/cups/fonts:/foo/bar/fonts
```

Description

The `FontPath` directive specifies the font path to use when searching for fonts. The default font path is `/usr/share/cups/fonts`.

Group

Examples

```
Group sys
Group system
Group root
```

Description

The `Group` directive specifies the UNIX group that filter and CGI programs run as. The default group is `sys`, `system`, or `root` depending on the operating system.

HideImplicitMembers

Examples

```
HideImplicitMembers Yes
HideImplicitMembers No
```

Description

The `HideImplicitMembers` directive controls whether the individual printers in an implicit class are shown to the user. The default is `No`.

<u>ImplicitClasses</u> must be enabled for this directive to have any effect.

HostNameLookups

Examples

```
HostNameLookups On
HostNameLookups Off
HostNameLookups Double
```

Description

The `HostNameLookups` directive controls whether or not CUPS looks up the hostname for connecting clients. The `Double` setting causes CUPS to verify that the hostname resolved from the address matches one of the addresses returned for that hostname. `Double` lookups also prevent clients with unregistered addresses from connecting to your server. The default is `Off` to avoid the potential server performance problems with hostname lookups. Set this option to `On` or `Double` only if absolutely required.

ImplicitClasses

Examples

```
ImplicitClasses On
ImplicitClasses Off
```

Description

The `ImplicitClasses` directive controls whether implicit classes are created based upon the available network printers and classes. The default setting is `On` but is automatically turned `Off` if <u>Browsing</u> is turned `Off`.

ImplicitAnyClasses

Examples

```
ImplicitAnyClasses On
ImplicitAnyClasses Off
```

Description

The `ImplicitAnyClasses` directive controls whether implicit classes for local and remote printers are created with the name `AnyPrinter`. The default setting is `Off`.

<u>ImplicitClasses</u> must be enabled for this directive to have any effect.

Include

Examples

```
Include filename
Include /foo/bar/filename
```

Description

The Include directive includes the named file in the cupsd.conf file. If no leading path is provided, the file is assumed to be relative to the ServerRoot directory.

KeepAlive

Examples

```
KeepAlive On
KeepAlive Off
```

Description

The KeepAlive directive controls whether or not to support persistent HTTP connections. The default is On.

HTTP/1.1 clients automatically support persistent connections, while HTTP/1.0 clients must specifically request them using the Keep-Alive attribute in the Connection: field of each request.

KeepAliveTimeout

Examples

```
KeepAliveTimeout 60
KeepAliveTimeout 30
```

Description

The KeepAliveTimeout directive controls how long a persistent HTTP connection will remain open after the last request. The default is 60 seconds.

Limit

Examples

```
<Limit GET POST>
...
</Limit>

<Limit ALL>
...
</Limit>
```

Description

The Limit directive groups access control directives for specific types of HTTP requests and must appear inside a <u>Location</u> section. Access can be limited for individual request types (DELETE, GET, HEAD, OPTIONS, POST, PUT, and TRACE) or for all request types (ALL). The request type names are case−sensitive for compatibility with Apache.

LimitExcept

Examples

```
<LimitExcept GET POST>
...
</LimitExcept>
```

Description

The LimitExcept directive groups access control directives for specific types of HTTP requests and must appear inside a <u>Location</u> section. Unlike the <u>Limit</u> directive, LimitExcept restricts access for all requests *except* those listed on the LimitExcept line.

LimitRequestBody

Examples

```
LimitRequestBody 10485760
LimitRequestBody 10m
LimitRequestBody 0
```

Description

The LimitRequestBody directive controls the maximum size of print files, IPP requests, and HTML form data in HTTP POST requests. The default limit is 0 which disables the limit check.

Also see the identical <u>MaxRequestSize</u> directive.

Listen

Examples

```
Listen 127.0.0.1:631
Listen 192.0.2.1:631
```

Description

The Listen directive specifies a network address and port to listen for connections. Multiple Listen directives can be provided to listen on multiple addresses.

The Listen directive is similar to the Port directive but allows you to restrict access to specific interfaces or networks.

Location

Examples

```
<Location />
...
</Location>

<Location /admin>
...
</Location>

<Location /printers>
...
</Location>

<Location /printers/name>
...
</Location>

<Location /classes>
...
</Location>

<Location /classes/name>
...
</Location>
```

Description

The Location directive specifies access control and authentication options for the specified HTTP resource or path. The Allow, AuthClass, AuthGroupName, AuthType, Deny, Encryption, Limit, LimitExcept, Order, Require, and Satisfy directives may all appear inside a location.

Locations on the Server.

Location	Description
/	The path for all get operations (get–printers, get–jobs, etc.)
/admin	The path for all administration operations (add–printer, delete–printer, start–printer, etc.)

/admin/conf	The path for access to the CUPS configuration files (cupsd.conf, client.conf, etc.)
/classes	The path for all classes
/classes/name	The resource for class name
/jobs	The path for all jobs (hold–job, release–job, etc.)
/jobs/id	The resource for job id
/printers	The path for all printers
/printers/name	The path for printer name
/printers/name.ppd	The PPD file path for printer name

Note that more specific resources override the less specific ones. So the directives inside the /printers/name location will override ones from /printers. Directives inside /printers will override ones from /. None of the directives are inherited. More information can be found in section "Printing System Security".

LogFilePerm

Examples

```
LogFilePerm 0644
LogFilePerm 0600
```

Description

The LogFilePerm directive specifies the permissions to use when writing configuration files. The default is 0644.

LogLevel

Examples

```
LogLevel none
LogLevel emerg
LogLevel alert
LogLevel crit
LogLevel error
LogLevel warn
LogLevel notice
LogLevel info
LogLevel debug
LogLevel debug2
```

Description

The LogLevel directive specifies the level of logging for the ErrorLog file. The following values are recognized (each level logs everything under the preceding levels):

- none – Log nothing.
- emerg – Log emergency conditions that prevent the server from running.
- alert – Log alerts that must be handled immediately.
- crit – Log critical errors that don't prevent the server from running.

- error – Log general errors.
- warn – Log errors and warnings.
- notice – Log temporary error conditions.
- info – Log all requests and state changes (default).
- debug – Log basic debugging information.
- debug2 – Log all debugging information.

MaxClients

Examples

```
MaxClients 100
MaxClients 1024
```

Description

The MaxClients directive controls the maximum number of simultaneous clients that will be allowed by the server. The default is 100 clients.

> **NOTE:**
>
> Since each print job requires a file descriptor for the status pipe, the CUPS server internally limits the MaxClients value to 1/3 of the available file descriptors to avoid possible problems when printing large numbers of jobs.

MaxClientsPerHost

Examples

```
MaxClientsPerHost 0
MaxClientsPerHost 10
```

Description

The MaxClientsPerHost directive controls the maximum number of simultaneous clients that will be allowed from a single host by the server. The default is 10 or 1/10th the MaxClients value, whichever is greater. A value of 0 uses the automatic setting based on the MaxClients value.

This directive provides a small measure of protection against Denial of Service attacks from a single host.

MaxCopies

Examples

```
MaxCopies 100
MaxCopies 65535
```

Description

The `MaxCopies` directive controls the maximum number of copies that a user can print of a job. The default is 100 copies.

> **NOTE:**
>
> Most HP PCL laser printers internally limit the number of copies to 100.

MaxJobs

Examples

```
MaxJobs 100
MaxJobs 9999
MaxJobs 0
```

Description

The `MaxJobs` directive controls the maximum number of jobs that are kept in memory. Once the number of jobs reaches the limit, the oldest completed job is automatically purged from the system to make room for the new one. If all of the known jobs are still pending or active then the new job will be rejected.

Setting the maximum to 0 disables this functionality. The default setting is 0.

MaxJobsPerPrinter

Examples

```
MaxJobsPerPrinter 100
MaxJobsPerPrinter 9999
MaxJobsPerPrinter 0
```

Description

The `MaxJobsPerPrinter` directive controls the maximum number of active jobs that are allowed for each printer or class. Once a printer or class reaches the limit, new jobs will be rejected until one of the active jobs is completed, stopped, aborted, or cancelled.

Setting the maximum to 0 disables this functionality. The default setting is 0.

MaxJobsPerUser

Examples

```
MaxJobsPerUser 100
MaxJobsPerUser 9999
MaxJobsPerUser 0
```

Description

The `MaxJobsPerUser` directive controls the maximum number of active jobs that are allowed for each user. Once a user reaches the limit, new jobs will be rejected until one of the active jobs is completed, stopped, aborted, or cancelled.

Setting the maximum to 0 disables this functionality. The default setting is 0.

MaxLogSize

Examples

```
MaxLogSize 1048576
MaxLogSize 1m
MaxLogSize 0
```

Description

The `MaxLogSize` directive controls the maximum size of each log file. Once a log file reaches or exceeds the maximum size it is closed and renamed to *filename.O*. This allows you to rotate the logs automatically. The default size is 1048576 bytes (1MB).

Setting the maximum size to 0 disables log rotation.

MaxRequestSize

Examples

```
MaxRequestSize 10485760
MaxRequestSize 10m
MaxRequestSize 0
```

Description

The `MaxRequestSize` directive controls the maximum size of print files, IPP requests, and HTML form data in HTTP POST requests. The default limit is 0 which disables the limit check.

Also see the identical <u>LimitRequestBody</u> directive.

Order

Examples

```
Order Allow,Deny
Order Deny,Allow
```

Description

The Order directive defines the default access control. The following values are supported:

- Allow,Deny – Allow requests from all systems *except* for those listed in a Deny directive.
- Deny,Allow – Allow requests only from those listed in an Allow directive.

The Order directive must appear inside a <u>Location</u> directive.

PageLog

Examples

```
PageLog /var/log/cups/page_log
PageLog /var/log/cups/page_log-%s
PageLog syslog
```

Description

The PageLog directive sets the name of the page log file. If the filename is not absolute then it is assumed to be relative to the <u>ServerRoot</u> directory. The default page log file is */var/log/cups/page_log*.

The server name can be included in the filename by using %s in the name.

The special name "syslog" can be used to send the page information to the system log instead of a plain file.

Port

Examples

```
Port 631
Port 80
```

Description

The Port directive specifies a port to listen on. Multiple Port lines can be specified to listen on multiple ports. The default port is 631.

PreserveJobHistory

Examples

```
PreserveJobHistory On
PreserveJobHistory Off
```

Description

The `PreserveJobHistory` directive controls whether the history of completed, cancelled, or aborted print jobs is stored on disk.

A value of `On` (the default) preserves job information until the administrator purges it with the `cancel` command.

A value of `Off` removes the job information as soon as each job is completed, cancelled, or aborted.

PreserveJobFiles

Examples

```
PreserveJobFiles On
PreserveJobFiles Off
```

Description

The `PreserveJobFiles` directive controls whether the document files of completed, cancelled, or aborted print jobs are stored on disk.

A value of `On` preserves job files until the administrator purges them with the `cancel` command. Jobs can be restarted (and reprinted) as desired until they are purged.

A value of `Off` (the default) removes the job files as soon as each job is completed, cancelled, or aborted.

Printcap

Examples

```
Printcap
Printcap /etc/printcap
Printcap /etc/printers.conf
```

Description

The `Printcap` directive controls whether or not a printcap file is automatically generated and updated with a list of available printers. If specified with no value, then no printcap file will be generated. The default is to generate a file named */etc/printcap*.

When a filename is specified (e.g. */etc/printcap*), the printcap file is written whenever a printer is added or removed. The printcap file can then be used by applications that are hardcoded to look at the printcap file for the available printers.

PrintcapFormat

Examples

```
PrintcapFormat BSD
PrintcapFormat Solaris
```

Description

The `PrintcapFormat` directive controls the output format of the printcap file. The default is to generate a BSD printcap file.

PrintcapGUI

Example

```
PrintcapGUI /usr/bin/glpoptions
```

Description

The `PrintcapGUI` directive sets the program to use when displaying an option panel from an IRIX application that uses the Impressario print API. The default program is the ESP Print Pro "glpoptions" GUI.

The program must accept the −d option to specify a printer and the −o option to specify one or more options. After allowing the user to select/change options, the program must then write the list of printing options without the −o to the standard output.

RemoteRoot

Examples

```
RemoteRoot remroot
RemoteRoot root
```

Description

The `RemoteRoot` directive sets the username for unauthenticated root requests from remote hosts. The default username is *remroot*. Setting `RemoteRoot` to *root* effectively disables this security mechanism.

RequestRoot

Examples

```
RequestRoot /var/spool/cups
RequestRoot /foo/bar/spool/cups
```

Description

The RequestRoot directive sets the directory for incoming IPP requests and HTML forms. If an absolute path is not provided then it is assumed to be relative to the ServerRoot directory. The default request directory is */var/spool/cups.*

Require

Examples

```
Require group foo bar
Require user john mary
Require valid-user
```

Description

The Require directive specifies that authentication is required for the resource. The group keyword specifies that the authenticated user must be a member of one or more of the named groups that follow.

The user keyboard specifies that the authenticated user must be one of the named users that follow.

The valid-user keyword specifies that any authenticated user may access the resource.

The default is to do no authentication. This directive must appear inside a Location directive.

RIPCache

Examples

```
RIPCache 8m
RIPCache 1g
RIPCache 2048k
```

Description

The RIPCache directive sets the size of the memory cache used by Raster Image Processor ("RIP") filters such as imagetoraster and pstoraster. The size can be suffixed with a "k" for kilobytes, "m" for megabytes, or "g" for gigabytes. The default cache size is "8m", or 8 megabytes.

RootCertDuration

Examples

```
RootCertDuration 300
RootCertDuration 0
```

Description

The `RootCertDuration` directive controls the interval between updates of the root authentication certificate. The default is `300` seconds which updates the root certificate approximately once every 5 minutes. Set the interval to 0 to disable certificate updates entirely.

RunAsUser

Examples

```
RunAsUser Yes
RunAsUser No
```

Description

The `RunAsUser` directive controls whether the scheduler runs as the unpriviledged user account (usually `lp`). The default is `No` which leaves the scheduler running as the `root` user.

Note: Running as a non–priviledged user may prevent LPD and locally connected printers from working due to permission problems. The `lpd` backend will automatically use a non–priviledged mode that is not 100% compliant with RFC 1179. The `parallel`, `serial`, and `usb` backends will need write access to the corresponding device files.

Satisfy

Examples

```
Satisfy all
Satisfy any
```

Description

The `Satisfy` directive specifies whether all conditions must be satisfied to allow access to the resource. If set to `all`, then all authentication and access control conditions must be satified to allow access.

Setting `Satisfy` to `any` allows a user to gain access if the authentication or access control requirements are satisfied. For example, you might require authentication for remote access, but allow local access without authentication.

The default is `all`. This directive must appear inside a <u>Location</u> directive.

ServerAdmin

Examples

```
ServerAdmin user@host
ServerAdmin root@foo.bar.com
```

Description

The ServerAdmin directive identifies the email address for the administrator on the system. By default the administrator email address is root@server, where server is the server name.

ServerBin

Examples

```
ServerBin /usr/lib/cups
ServerBin /foo/bar/lib/cups
```

Description

The ServerBin directive sets the directory for server−run executables. If an absolute path is not provided then it is assumed to be relative to the ServerRoot directory. The default executable directory is */usr/lib/cups.*

ServerCertificate

Examples

```
ServerCertificate /etc/cups/ssl/server.crt
```

Description

The ServerCertificate directive specifies the location of the SSL certificate file used by the server when negotiating encrypted connections. The certificate must not be encrypted (password protected) since the scheduler normally runs in the background and will be unable to ask for a password. The default certificate file is */etc/cups/ssl/server.crt.*

ServerKey

Examples

```
ServerKey /etc/cups/ssl/server.key
```

Description

The `ServerKey` directive specifies the location of the SSL private key file used by the server when negotiating encrypted connections. The default key file is */etc/cups/ssl/server.crt*.

ServerName

Examples

```
ServerName foo.domain.com
ServerName myserver.domain.com
```

Description

The `ServerName` directive specifies the hostname that is reported to clients. By default the server name is the hostname.

ServerRoot

Examples

```
ServerRoot /etc/cups
ServerRoot /foo/bar/cups
```

Description

The `ServerRoot` directive specifies the absolute path to the server configuration and state files. It is also used to resolve relative paths in the *cupsd.conf* file. The default server directory is */etc/cups*.

SSLListen

Examples

```
SSLListen 127.0.0.1:443
SSLListen 192.0.2.1:443
```

Description

The `SSLListen` directive specifies a network address and port to listen for secure connections. Multiple `SSLListen` directives can be provided to listen on multiple addresses.

The `SSLListen` directive is similar to the SSLPort directive but allows you to restrict access to specific interfaces or networks.

SSLPort

Examples

```
SSLPort 443
```

Description

The `SSLPort` directive specifies a port to listen on for secure connections. Multiple `SSLPort` lines can be specified to listen on multiple ports.

SystemGroup

Examples

```
SystemGroup sys
SystemGroup system
SystemGroup root
```

Description

The `SystemGroup` directive specifies the system administration group for `System` authentication. More information can be found later in this chapter in "Printing System Security".

TempDir

Examples

```
TempDir /var/tmp
TempDir /foo/bar/tmp
```

Description

The `TempDir` directive specifies an absolute path for the directory to use for temporary files. The default directory is */var/tmp.*

Temporary directories must be world−writable and should have the "sticky" permission bit enabled so that other users cannot delete filter temporary files. The following commands will create an appropriate temporary directory called */foo/bar/tmp*:

```
mkdir /foo/bar/tmp ENTER
chmod a+rwxt /foo/bar/tmp ENTER
```

Timeout

Examples

```
Timeout 300
Timeout 90
```

Description

The `Timeout` directive controls the amount of time to wait before an active HTTP or IPP request times out. The default timeout is 300 seconds.

User

Examples

```
User lp
User guest
```

Description

The `User` directive specifies the UNIX user that filter and CGI programs run as. The default user is `lp`.

Changing the Client Configuration

The CUPS client application (`lp`, `lpr`, and so forth) use the */etc/cups/client.conf* file for default settings. The client application also look in the user's home directory for a file called *.cupsrc*. Each directive is listed on a line by itself followed by its value. Comments are introduced using the number sign ("#") character at the beginning of a line.

Since the client configuration file consists of plain text, you can use your favorite text editor to make changes to it.

Client Directives

The *client.conf* file contains many directives that determine how the client behaves:

- <u>Encryption</u>
- <u>ServerName</u>

Encryption

Examples

```
Encryption Never
Encryption IfRequested
Encryption Required
Encryption Always
```

Description

The `Encryption` directive specifies the default encryption settings for the client. The default setting is *IfRequested*.

ServerName

Examples

```
ServerName foo.bar.com
ServerName 11.22.33.44
```

Description

The `ServerName` directive specifies sets the remote server that is to be used for all client operations. That is, it redirects all client requests to the remote server. The default is to use the local server ("*localhost*").

Changing the Printer Configuration

The CUPS scheduler (cupsd) uses the */etc/cups/printers.conf* file to store the list of available printers. This file contains only locally defined printers, but not remote printers that are created automatically. Each directive is listed on a line by itself followed by its value. Comments are introduced using the number sign ("#") character at the beginning of a line.

Since the printer configuration file consists of plain text, you can use your favorite text editor to make changes to it.

Printer Directives

The *printers.conf* file contains many directives that determine how the printer behaves:

- Accepting
- AllowUser
- DefaultPrinter
- DenyUser
- DeviceURI

- Info
- JobSheets
- KLimit
- Location
- PageLimit

- Printer
- QuotaPeriod
- State
- StateMessage

Accepting

Examples

```
Accepting yes

Accepting no
```

Description

The Accepting directive defines the initial Boolean value for the *printer–is–accepting–job* attribute which can be set by the accept and reject commands.

This directive must appear inside a Printer or DefaultPrinter directive.

AllowUser

Examples

```
AllowUser foo_user
AllowUser bar_user
```

Description

The AllowUser directive adds a username to the *requesting–user–name–allowed* attribute which can be set by the lpadmin –u command.

This directive must appear inside a Printer or DefaultPrinter directive.

DefaultPrinter

Examples

```
<DefaultPrinter name/>
...
</Printer>
```

Description

The DefaultPrinter directive begins a printer definition for the default server destination. It can be added by the lpadmin command or if already defined, set as default by the lpoptions −d command.

DenyUser

Examples

```
DenyUser foo_user
DenyUser bar_user
```

Description

The DenyUser directive adds a username to the *requesting−user−name−allowed* attribute which can be set by the lpadmin −u command.

This directive must appear inside a <u>Printer</u> or <u>DefaultPrinter</u> directive.

DeviceURI

Examples

```
DeviceURI socket://foo.bar.com:9100
```

Description

The DeviceURI directive defines the value of the *device−uri−attribute* attribute which can be set by the lpadmin −v command.

This directive must appear inside a <u>Printer</u> or <u>DefaultPrinter</u> directive.

Info

Examples

```
Info My Printer
```

Description

The `Info` directive defines the string for the *printer–info* attribute which can be set by the `lpadmin -D` command.

This directive must appear inside a <u>Printer</u> or <u>DefaultPrinter</u> directive.

JobSheets

Examples

```
JobSheets none,standard
```

Description

The `JobSheets` directive specifies the default banner pages to print before and after a print job. In the above example, only a *standard* banner will print after each job.

The `lpoptions -o job-sheets=` command can be used to set banners. For example, the following command would produce the same results of a *standard* banner at the end of each print job for the default printer.

lpoptions –o job–sheets=none,standard

If only one banner file is specified, it will be printed before the files in the job. If a second banner file is specified, it is printed after the files in the job.

The available banner pages depend on the local system configuration; CUPS includes the following banner files:

- `none` – Do not produce a banner page.
- `classified` – A banner page with a "classified" label at the top and bottom.
- `confidential` – A banner page with a "confidential" label at the top and bottom.
- `secret` – A banner page with a "secret" label at the top and bottom.
- `standard` – A banner page with no label at the top and bottom.
- `topsecret` – A banner page with a "top secret" label at the top and bottom.
- `unclassified` – A banner page with an "unclassified" label at the top and bottom.

This directive must appear inside a <u>Printer</u> or <u>DefaultPrinter</u> directive.

KLimit

Examples

```
KLimit 1234
```

Description

The `KLimit` directive defines the value of the *job−k−limit* attribute which can be set by the `lpadmin −o job-k-limit=` command.

This directive must appear inside a <u>Printer</u> or <u>DefaultPrinter</u> directive.

Location

Examples

```
Location Building 3321
```

Description

The `Location` directive defines the string for the *printer−location* attribute which can be set by the `lpadmin −L` command.

> **NOTE:**
>
> Do not confuse this `Location` directive with the one for cupsd.conf. They are completely different.

This directive must appear inside a <u>Printer</u> or <u>DefaultPrinter</u> directive.

PageLimit

Examples

```
PageLimit 1234
```

Description

The `PageLimit` directive defines the value of the *job−page−limit* attribute which can be set by the `lpadmin -o job-page-limit=` command.

This directive must appear inside a <u>Printer</u> or <u>DefaultPrinter</u> directive.

Printer

Examples

```
<Printer name/>
...
</Printer>
```

Description

The `Printer` directive begins a printer definition. It can be added by the `lpadmin` command.

QuotaPeriod

Examples

```
QuotaPeriod 604800
```

Description

The `QuotaPeriod` directive defines the value of the *job−quota−period* attribute which can be set by the `lpadmin -o job-quota-period=` command.

This directive must appear inside a <u>Printer</u> or <u>DefaultPrinter</u> directive.

State

Examples

```
State stopped
```

Description

The `State` directive defines the initial value of the *printer−state* attribute. The strings *idle* and *stopped* correspond to the IPP enumeration values.

This directive must appear inside a <u>Printer</u> or <u>DefaultPrinter</u> directive.

StateMessage

Examples

```
StateMessage Ready to print.
```

Description

The `StateMessage` directive defines the initial string for the *printer–state–message* attribute. The following are some example messages:

- Connected to *host_name*...
- Connecting to *printer_queue* on port *port_number*...
- Network host *host_name* is busy; will retry in 30 seconds...
- Printer busy; will retry in 10 seconds...
- Printer is busy; retrying print job...
- Print file accepted – job ID *id_number*.
- Ready to print.
- Waiting for job to complete

This directive must appear inside a <u>Printer</u> or <u>DefaultPrinter</u> directive.

Changing the Classes Configuration

The CUPS scheduler (cupsd) uses the */etc/cups/classes.conf* file to store the list of available classes. This file contains only locally defined classes, but not remote or implicit classes that are created automatically. Each directive is listed on a line by itself followed by its value. Comments are introduced using the number sign ("#") character at the beginning of a line.

Since the classes configuration file consists of plain text, you can use your favorite text editor to make changes to it.

Classes Directives

The *classes.conf* file contains many directives that determine how the classes behaves:

- <u>Accepting</u>
- <u>AllowUser</u>
- <u>Class</u>
- <u>DefaultClass</u>
- <u>DenyUser</u>
- <u>Info</u>
- <u>JobSheets</u>
- <u>KLimit</u>
- <u>Location</u>
- <u>PageLimit</u>
- <u>Printer</u>
- <u>QuotaPeriod</u>
- <u>State</u>
- <u>StateMessage</u>

Accepting

Examples

```
Accepting yes

Accepting no
```

Description

The `Accepting` directive defines the initial Boolean value for the *printer–is–accepting–job* attribute which can be set by the `accept` and `reject` commands.

This directive must appear inside a <u>Class</u> or <u>DefaultClass</u> directive.

AllowUser

Examples

```
AllowUser foo_user
AllowUser bar_user
```

Description

The `AllowUser` directive adds a username to the *requesting–user–name–allowed* attribute which can be set by the `lpadmin -u` command.

This directive must appear inside a <u>Class</u> or <u>DefaultClass</u> directive.

Class

Examples

```
<Class name/>
...
</Class>
```

Description

The `Class` directive begins a class definition. It can be added by the lpadmin −c command.

DefaultClass

Examples

```
<DefaultClass name/>
...
</DefaultClass>
```

Description

The `DefaultClass` directive begins a class definition for the default server destination. It can be added by the lpadmin −c command or if already defined, set as default by the lpoptions −d command.

DenyUser

Examples

```
DenyUser foo_user
DenyUser bar_user
```

Description

The `DenyUser` directive adds a username to the *requesting−user−name−allowed* attribute which can be set by the lpadmin −u command.

This directive must appear inside a <u>Class</u> or <u>DefaultClass</u> directive.

Info

Examples

```
Info My Printer
```

Description

The `Info` directive defines the string for the *printer–info* attribute which can be set by the `lpadmin -D` command.

This directive must appear inside a <u>Class</u> or <u>DefaultClass</u> directive.

JobSheets

Examples

```
JobSheets none,standard
```

Description

The `JobSheets` directive specifies the default banner pages to print before and after a print job. In the above example, only a *standard* banner will print after each job.

The `lpoptions -o job-sheets=` command can be used to set banners. For example, the following command would produce the same results of a *standard* banner at the end of each print job for the default class.

lpoptions –o job–sheets=none,standard

If only one banner file is specified, it will be printed before the files in the job. If a second banner file is specified, it is printed after the files in the job.

The available banner pages depend on the local system configuration; CUPS includes the following banner files:

- `none` – Do not produce a banner page.
- `classified` – A banner page with a "classified" label at the top and bottom.
- `confidential` – A banner page with a "confidential" label at the top and bottom.
- `secret` – A banner page with a "secret" label at the top and bottom.
- `standard` – A banner page with no label at the top and bottom.
- `topsecret` – A banner page with a "top secret" label at the top and bottom.
- `unclassified` – A banner page with an "unclassified" label at the top and bottom.

This directive must appear inside a <u>Class</u> or <u>DefaultClass</u> directive.

KLimit

Examples

```
KLimit 1234
```

Description

The `KLimit` directive defines the value of the *job-k-limit* attribute which can be set by the `lpadmin -o job-k-limit=` command.

This directive must appear inside a <u>Class</u> or <u>DefaultClass</u> directive.

Location

Examples

```
Location Building 3321
```

Description

The `Location` directive defines the string for the *printer-location* attribute which can be set by the `lpadmin -L` command.

> **NOTE:**
>
> Do not confuse this `Location` directive with the one for cupsd.conf. They are completely different.

This directive must appear inside a <u>Class</u> or <u>DefaultClass</u> directive.

PageLimit

Examples

```
PageLimit 1234
```

Description

The `PageLimit` directive defines the value of the *job-page-limit* attribute which can be set by the `lpadmin -o job-page-limit=` command.

This directive must appear inside a <u>Class</u> or <u>DefaultClass</u> directive.

Printer

Examples

```
Printer foo
Printer bar
Printer foo@bar
```

Description

The `Printer` directive adds a printer to the class. It can be added by the `lpadmin -c` command.

> **NOTE:**
>
> Do not confuse this `Printer` directive with the one for printers.conf. They are completely different.

This directive must appear inside a <u>Class</u> or <u>DefaultClass</u> directive.

QuotaPeriod

Examples

```
QuotaPeriod 604800
```

Description

The `QuotaPeriod` directive defines the value of the *job–quota–period* attribute which can be set by the `lpadmin -o job-quota-period=` command.

This directive must appear inside a <u>Class</u> or <u>DefaultClass</u> directive.

State

Examples

```
State stopped
```

Description

The `State` directive defines the initial value of the *printer–state* attribute. The strings *idle* and *stopped* correspond to the IPP enumeration values.

This directive must appear inside a <u>Class</u> or <u>DefaultClass</u> directive.

StateMessage

Examples

```
StateMessage Ready to print.
```

Description

The `StateMessage` directive defines the initial string for the *printer−state−message* attribute. The following are some example messages:

- Connected to *host_name*...
- Connecting to *class_queue* on port *port_number*...
- Network host *host_name* is busy; will retry in 30 seconds...
- Printer busy; will retry in 10 seconds...
- Printer is busy; retrying print job...
- Print file accepted – job ID *id_number*.
- Ready to print.
- Waiting for job to complete

This directive must appear inside a <u>Class</u> or <u>DefaultClass</u> directive.

Printing System Security

CUPS provides support for address, certificate, and password (Basic and Digest) based authentication and access control. Certificate and password authentication provide ways to limit access to individual people or groups.

Address based access control allows you to limit access to specific systems, networks, or domains. While this does not provide authentication, it does allow you to limit the potential users of your system efficiently.

CUPS maintains a list of locations that have access control and/or authentication enabled. Locations are specified using the Location directive:

```
<Location /resource>
AuthClass ...
AuthGroupName ...
AuthType ...

Order ...
Allow from ...
Deny from ...
</Location>
```

Locations generally follow the directory structure of the DocumentRoot directory, however CUPS does have several virtual locations for administration, classes, jobs, and printers:

Location	Description
/admin	The path for all administration operations.
/classes	The path for all classes.
/classes/name	The resource for class name.
/jobs	The path for all jobs.
/jobs/id	The resource for job id.
/printers	The path for all printers.
/printers/name	The path for printer name.
/printers/name.ppd	The PPD file path for printer name.

Authentication Using Certificates

CUPS supports a local certificate–based authentication scheme that can be used in place of Basic or Digest authentication by clients connecting through the localhost interface. Certificate authentication is not supported or allowed from clients on any other interface.

Certificates are 128–bit random numbers that refer to an internal authentication record in the server. A client connecting via the localhost interface sends a request with an authorization header of:

```
Authorization: Local 0123456789ABCDEF0123456789ABCDEF
```

The server then looks up the local certificate and authenticates using the username associated with it.

Certificates are generated by the server automatically and stored in the */etc/cups/certs* directory using the process ID of the CGI program started by the server. Certificate files are only readable by the User and

Group defined in the *cupsd.conf* file. When the CGI program ends the certificate is removed and invalidated automatically.

The special file */etc/cups/certs/0* defines the *root certificate* which can be used by any client running as the super−user or another user that is part of the group defined by the SystemGroup directive. The root certificate is automatically regenerated every 5 minutes.

Using Basic Authentication

Basic authentication uses UNIX users and passwords to authenticate access to resources such as printers and classes, and to limit access to administrative functions.

> **NOTE:**
>
> Basic authentication sends the username and password Base64 encoded from the client to the server, so it offers no protection against eavesdropping. This means that a malicious user can monitor network packets and discover valid users and passwords that could result in a serious compromise in network security. Use Basic authentication with extreme care.

The CUPS implementation of Basic authentication does not allow access through user accounts without a password. If you try to authenticate using an account without a password, your access will be immediately blocked.

Once a valid username and password is authenticated by CUPS, any additional group membership requirements are checked.

> **NOTE:**
>
> The root user is considered by CUPS to be a member of every group.

Use the AuthType directive to enable Basic authentication:

```
AuthType Basic
```

Using Digest Authentication

Digest authentication uses users and passwords defined in the */etc/cups/passwd.md5* file to authenticate access to resources such as printers and classes, and to limit access to administrative functions.

> **NOTE:**
>
> Unlike Basic authentication, Digest passes the MD5 sum (basically a complicated checksum) of the username and password instead of the strings themselves. Also, Digest authentication does not use the UNIX password file, so if an attacker does discover the original password it is less likely to result in a serious security problem so long as you use a different UNIX password than the corresponding Digest password.
>
> The current CUPS implementation of Digest authentication uses the client's hostname or IP address for the "nonce" value. The nonce value is an additional string added to the username and password to make guessing the password more difficult. The server checks that the nonce value matches the client's hostname or address and rejects the MD5 sum if it doesn't. Future versions of CUPS will support Digest "session" authentication which adds the request data to the MD5 sum, providing even better authentication and security.
>
> Digest authentication does not guarantee that an attacker cannot gain unauthorized access, but it is safer than Basic authentication and should be used in place of Basic authentication whenever possible. **Support for Digest authentication in web browsers is not yet universally available.**

The lppasswd(1) command is used to add, change, or remove accounts from the *passwd.md5* file. To add a user to the default system group, type:

```
lppasswd -a user ENTER
Password: (password) ENTER [password is not echoed]
Password again: (password) ENTER [password is not echoed]
```

Once added, a user can change his/her password by typing:

```
lppasswd ENTER
Old password: (password) ENTER [password is not echoed]
Password: (password) ENTER [password is not echoed]
Password again: (password) ENTER [password is not echoed]
```

To remove a user from the password file, type:

```
lppasswd -x user ENTER
```

Once a valid username and password is authenticated by CUPS, any additional group membership requirements are checked.

> **NOTE:**
>
> The root user is considered by CUPS to be a member of every group.

Use the `AuthType` directive to enable Digest authentication:

```
AuthType Digest
```

System and Group Authentication

The <u>AuthClass</u> directive controls the level of authentication to perform. `System` and `Group` authentication extend the normal user–based authentication to require membership in a UNIX group. For `System` authentication each user must belong to the `sys`, `system`, or `root` group; the actual group depends on the operating system.

For `Group` authentication each user must belong to the group named by the <u>AuthGroupName</u> directive:

```
<Location /path>
AuthType Digest
AuthClass Group
AuthGroupName mygroup
</Location>
```

The named group must be a valid UNIX user group, usually defined in the */etc/group* or */etc/netgroup* files. Additionally, when using Digest authentication you need to create user accounts with the named group:

```
lppasswd -g mygroup -a user ENTER
Password: (password) ENTER [password is not echoed]
Password again: (password) ENTER [password is not echoed]
```

Printer Accounting

CUPS maintains a log of all accesses, errors, and pages that are printed. The log files are normally stored in the */var/log/cups* directory. You can change this by editing the */etc/cups/cupsd.conf* configuration file.

The access_log File

The *access_log* file lists each HTTP resource that is accessed by a web browser or CUPS/IPP client. Each line is in the so–called "Common Log Format" used by many web servers and web reporting tools:

```
host group user date-time \"method resource version\" status bytes

127.0.0.1 - - [20/May/1999:19:20:29 +0000] "POST /admin/ HTTP/1.1" 401 0
127.0.0.1 - mike [20/May/1999:19:20:31 +0000] "POST /admin/ HTTP/1.1" 200 0
```

The *host* field will normally only be an IP address unless you have enabled the <u>HostNameLookups</u> directive in the *cupsd.conf* file.

The *group* field always contains "–" in CUPS.

The *user* field is the authenticated username of the requesting user. If no username and password is supplied for the request then this field contains "–".

The *date–time* field is the date and time of the request in local time and is in the format:

```
[DD/MON/YYYY:HH:MM:SS +ZZZZ]
```

where *ZZZZ* is the timezone offset in hours and minutes from coordinated universal time (UTC). UTC may sometimes be referred to as GMT or ZULU on legacy systems.

The *method* field is the HTTP method used ("GET", "PUT", "POST", etc.)

The *resource* field is the filename of the requested resource.

The *version* field is the HTTP specification version used by the client. For CUPS clients this will always be "HTTP/1.1".

The *status* field contains the HTTP result status of the request. Usually it is "200", but other HTTP status codes are possible. For example, 401 is the "unauthorized access" status in the example above.

The *bytes* field contains the number of bytes in the request. For POST requests the *bytes* field contains the number of bytes that was received from the client.

The error_log File

The *error_log* file lists messages from the scheduler (errors, warnings, etc.):

```
level date-time message

I [20/May/1999:19:18:28 +0000] Job 1 queued on 'DeskJet' by 'mike'.
I [20/May/1999:19:21:02 +0000] Job 2 queued on 'DeskJet' by 'mike'.
I [20/May/1999:19:22:24 +0000] Job 2 was cancelled by 'mike'.
```

The *level* field contains the type of message:

- `E` – An error occurred.
- `W` – The server was unable to perform some action.
- `I` – Informational message.
- `D` – Debugging message.

The *date−time* field contains the date and time of when the page started printing. The format of this field is identical to the *data−time* field in the **access_log** file.

The *message* fields contains a free−form textual message.

The page_log File

The **page_log** file lists each page that is sent to a printer. Each line contains the following information:

```
printer user job-id date-time page-number num-copies job-billing

DeskJet root 2 [20/May/1999:19:21:05 +0000] 1 0 acme-123
```

The *printer* field contains the name of the printer that printed the page. If you send a job to a printer class, this field will contain the name of the printer that was assigned the job.

The *user* field contains the name of the user (the IPP `requesting-user-name` attribute) that submitted this file for printing.

The *job−id* field contains the job number of the page being printed. Job numbers are reset to 1 whenever the CUPS server is started, so don't depend on this number being unique!

The *date−time* field contains the date and time of when the page started printing. The format of this field is identical to the *data−time* field in the **access_log** file.

The *page−number* and *num−pages* fields contain the page number and number of copies being printed of that page. For printer that can not produce copies on their own, the *num−pages* field will always be 1.

The *job−billing* field contains a copy of the `job-billing` attribute provided with the IPP `create-job` or `print-job` requests or "−" if none was provided.

File Typing and Filtering

CUPS provides a MIME–based file typing and filtering mechanism to convert files to a printable format for each printer. On startup the CUPS server reads MIME database files from the */etc/cups* directory (or a directory specified by the <u>ServerRoot</u> directive) to build a file type and conversion database in memory. These database files are plain ASCII text and can be edited with your favorite text editor.

The *mime.types* and *mime.convs* files define the standard file types and filters that are available on the system.

mime.types

The *mime.types* file defines the known file types. Each line of the file starts with the MIME type and may be followed by one or more file type recognition rules. For example, the `text/html` file type is defined as:

```
text/html       html htm \
        printable(0,1024) + \
        (string(0,"<HTML>") string(0,"<!DOCTYPE"))
```

The first two rules say that any file with an extension of *.html* or *.htm* is a HTML file. The third rule says that any file whose first 1024 characters are printable text and starts with the strings `<HTML>` or `<!DOCTYPE` is a HTML file as well.

The first two rules deal solely with the name of the file being typed. This is useful when the original filename is known, however for print files the server doesn't have a filename to work with. The third rule takes care of this possibility and automatically figures out the file type based upon the contents of the file instead.

The available tests are:

- `( expr )` – Parenthesis for expression grouping
- `+` – Logical AND
- `,` or whitespace – Logical OR
- `!` – Logical NOT
- `match("pattern")` – Pattern match on filename
- `extension` – Pattern match on "*.extension"
- `ascii(offset,length)` – True if bytes are valid printable ASCII (CR, NL, TAB, BS, 32–126)
- `printable(offset,length)` – True if bytes are printable 8–bit chars (CR, NL, TAB, BS, 32–126, 160–254)
- `string(offset,"string")` – True if bytes are identical to string
- `contains(offset,range,"string")` – True if the range of bytes contains the string
- `char(offset,value)` – True if byte is identical
- `short(offset,value)` – True if 16–bit integer is identical (network or "big–endian" byte order)
- `int(offset,value)` – True if 32–bit integer is identical (network or "big–endian" byte order)
- `locale("string")` – True if current locale matches string

All numeric values can be in decimal (123), octal (0123), or hexadecimal (0x123) as desired.

Strings can be in quotes, all by themselves, as a string of hexadecimal values, or some combination:

```
"string"
'string'
string
<737472696e67>
<7374>ring
```

As shown in the `text/html` example, rules can continue on multiple lines using the backslash (\) character. A more complex example is the `image/jpeg` rules:

```
image/jpeg      jpeg jpg jpe string(0,<FFD8FF>) &&\
        (char(3,0xe0) char(3,0xe1) char(3,0xe2) char(3,0xe3)\
         char(3,0xe4) char(3,0xe5) char(3,0xe6) char(3,0xe7)\
         char(3,0xe8) char(3,0xe9) char(3,0xea) char(3,0xeb)\
         char(3,0xec) char(3,0xed) char(3,0xee) char(3,0xef))
```

This rule states that any file with an extension of *.jpeg*, *.jpg*, or *.jpe* is a JPEG file. In addition, any file starting with the hexadecimal string `<FFD8FF>` (JPEG Start–Of–Image) followed by a character between and including `0xe0` and `0xef` (JPEG APPn markers) is also a JPEG file.

mime.convs

The *mime.convs* file defines all of the filter programs that are known to the system. Each line consists of:

```
source destination cost program

text/plain application/postscript 50 texttops
application/vnd.cups-postscript application/vnd.cups-raster 50 pstoraster
image/* application/vnd.cups-postscript 50 imagetops
image/* application/vnd.cups-raster 50 imagetoraster
```

The *source* field is a MIME type, optionally using a wildcard for the super–type or sub–type (e.g. "text/plain", "image/*", "*/postscript").

The *destination* field is a MIME type defined in the **mime.types** file.

The *cost* field defines a relative cost for the filtering operation from 1 to 100. The cost is used to choose between two different sets of filters when converting a file. For example, to convert from `image/jpeg` to `application/vnd.cups-raster`, you could use the `imagetops` and `pstoraster` filters for a total cost of 100, or the `imagetoraster` filter for a total cost of 50.

The *program* field defines the filter program to run; the special program "–" can be used to make two file types equivalent. The program must accept the standard filter arguments and environment variables described in the CUPS Interface Design Description and CUPS Software Programmers Manual:

```
program job user title options [filename]
```

If specified, the *filename* argument defines a file to read when filtering, otherwise the filter must read from the standard input. All filtered output must go to the standard output.

Adding Filetypes and Filters

Adding a new file type or filter is fairly straight−forward. Rather than adding the new type and filter to the *mime.types* and *mime.convs* files which are overwritten when you upgrade to a new version of CUPS, you simple need to create new files with *.types* and *.convs* extensions in the */etc/cups* directory. We recommend that you use the product or format name, e.g.:

```
myproduct.types
myproduct.convs
```

If you are providing a filter for a common file format or printer, add the company or author name:

```
acme-msword.types
acme.msword.convs
```

This will help to prevent name collisions if you install many different file types and filters.

Once you choose the names for these files, create them using your favorite text editor as described earlier in this chapter. Once you have created the files, restart the `cupsd` process as described earlier in "Restarting the CUPS Server".

Printer Drivers and PPD Files

Most CUPS printer drivers utilize one or more printer−specific filters and a PPD file for each printer model. Printer driver filters are registered via the PPD file using `cupsFilter` attributes:

```
*cupsFilter: "application/vnd.cups-raster 0 rastertohp"
```

The filter is specified using the source file type only; the destination file type is assumed to be `printer/name` − suitable for sending to the printer.

Writing Your Own Filter or Printer Driver

CUPS supports an unlimited number of file formats and filters, and can handle any printer. If you'd like to write a filter or printer driver for your favorite file format or printer, consult the CUPS Software Programmers Manual for step−by−step instructions.

7 – Printing with Other Systems

This chapter describes how to print from client systems that use the LPD, Mac OS, or Windows printing protocols.

The Basics

CUPS is based on the IPP protocol, so any system that supports IPP can send jobs to and receive jobs from CUPS automatically. However, not all systems support IPP yet. This chapter will show you how to connect these systems to your CUPS server, either to accept jobs from your server for printing, or to send jobs to your server.

Printing from LPD Clients

CUPS supports limited functionality for LPD−based clients. With LPD you can print files to specific printers, list the queue status, and so forth. However, the automatic client configuration and printer options are not supported by the LPD protocol, so you must manually configure each client for the printers it needs to access.

The `cups-lpd(8)` program provides support for LPD clients and can be used from either the `inetd(8)` or `xinetd(8)` programs. Add the following line to the */etc/inetd.conf* file to enable LPD support on your server through the `inetd` program:

```
printer stream tcp nowait lp /usr/lib/cups/daemon/cups-lpd cups-lpd
```

The path to the `cups-lpd` may vary depending on your installation.

Once you have added this line, send the `inetd` process a `HUP` signal or reboot the system:

```
killall -HUP inetd ENTER [IRIX and some versions of Linux]
kill -HUP pid ENTER [Others]
reboot ENTER [For all systems if the HUP signal fails]
```

If you are using the `xinetd` program, create a file named */etc/xinetd.d/printer* containing the following lines:

```
service printer
{
socket_type = stream
protocol = tcp
wait = no
user = lp
server = /usr/lib/cups/daemon/cups-lpd
}
```

The `xinetd` program automatically reads the new configuration file and enables LPD printing support.

> **Warning:**
>
> `cups-lpd` currently does not perform any access control based on the settings in *cupsd.conf* or in the *hosts.allow* or *hosts.deny* files used by TCP wrappers. Therefore, running `cups-lpd` on your server will allow any computer on your network (and perhaps the entire Internet) to print to your server.
>
> While `xinetd` has built−in access control support, you should use the TCP wrappers package with `inetd` to limit access to only those computers that should be able to print through your server.

Printing to LPD Servers

CUPS provides the `lpd` backend for printing to LPD−based servers and printers. Use a device URI of `lpd://server/name` to print to a printer on an LPD server, where `server` is the hostname or IP address of the server and `name` is the queue name.

Microsoft Windows NT provides an LPD service under the name "TCP/IP Printing Services". To enable LPD printing on NT, open the "Services" control panel, select the "TCP/IP Printing Services" service, and click on the "Start" button. Any shared printer will then be available via the LPD protocol.

Printing from Mac OS Clients

CUPS does not provide Mac OS support directly. However, there are several free and commercial software packages that do.

Columbia Appletalk Package (CAP)

Because the CAP LaserWriter server (`lwsrv(8)`) does not support specification of PPD files, we do not recommend that you use CAP with CUPS. However, you can run the `lpsrv` program for limited printing with the command:

```
lwsrv −n "Name" −p printer −a /usr/lib/adicts −f /usr/lib/LW+Fonts
```

where `Name` is the name you want to use when sharing the printer, and `printer` is the name of the CUPS print queue.

XINET KA/Spool

To use your system as a print server for Mac OS clients, configure each printer using a `papserver(8)` in the */usr/adm/appletalk/services* file, specifying the corresponding PPD file in the */etc/cups/ppd* directory for each printer. For a printer named `MyPrinter` the entry would look like:

```
/usr/etc/appletalk/papserver −I −L −P /etc/cups/ppd/MyPrinter.ppd \
"Printer Description" MyPrinter
```

> **NOTE:**
>
> Enter the text above on a single line without the backslash (\) character.

NetATalk

To use your system as a print server for Mac OS clients, configure each printer in the *papd.conf* file, specifying the corresponding PPD file in the */etc/cups/ppd* directory for each printer. For a printer named `MyPrinter` the entry would look like:

```
Printer Description:MyPrinter@MyServer:\
:pr=|/usr/bin/lp −d MyPrinter:\
:op=daemon:\
:pd=/etc/cups/ppd/MyPrinter.ppd:
```

Printing to Mac OS Servers

CUPS currently does not provide a backend to communicate with a Mac OS server. However, you can write and install a short shell script in the */usr/lib/cups/backend* directory that sends a print file using the appropriate command. The following is a short script that will run the `papif` command provided with CAP.

After copying this script to */usr/lib/cups/backend/cap*, specify a device URI of `cap://server/printer` to use this backend with a print queue.

"/usr/lib/cups/backend/cap"

```
#!/bin/sh
#
# Usage: cap job user title copies options [filename]
#

# No arguments means show available devices...

if test ${#argv} = 0; then
echo "network cap \"Unknown\" \"Mac OS Printer via CAP\""
exit 0
fi

# Collect arguments...

user=$2
copies=$4

if test ${#argv} = 5; then
# Get print file from stdin; copies have already been handled...
file=/var/tmp/$$.prn
copies=1
cat > $file
else
# Print file is on command-line...
file=$6
fi

# Create a dummy cap.printers file for this printer based
# upon a device URI of "cap://server/printer"...

echo $PRINTER/$DEVICE_URI | \
awk -F/ '{print $1 "=" $5 ":LaserWriter@" $4}' > /var/tmp/$$.cap

CAPPRINTERS=/var/tmp/$$.cap; export CAPPRINTERS

# Send the file to the printer, once for each copy. This assumes that you
# have properly initialized the cap.printers file...

while [ $copies -gt 0 ]; do
papif -n $user < $file

copies=`expr $copies - 1`
done

# Remove any temporary files...
if test ${#argv} = 5; then
/bin/rm -f $file
fi

/bin/rm -f /var/tmp/$$.cap

exit 0
```

Printing from Windows Clients

While CUPS does not provide Windows support directly, the free SAMBA software package does. SAMBA version 2.0.6 is the first release of SAMBA that supports CUPS. You can download SAMBA from:

> http://www.samba.org

To configure SAMBA for CUPS, edit the *smb.conf* file and replace the existing printing commands and options with the line:

```
printing = cups
printcap name = cups
```

That's all there is to it! Remote users will now be able to browse and print to printers on your system.

Exporting Printer Drivers

You can optionally export printer drivers from your CUPS server using the `cupsaddsmb` command and the SAMBA 2.2.0 or higher software.

Before you can export the printers you must download the current Adobe PostScript printer drivers from the Adobe web site (http://www.adobe.com/). Use the free `unzip` software to extract the files from the self−extracting ZIP file containing the drivers; you will need the following files:

```
ADFONTS.MFM
ADOBEPS4.DRV
ADOBEPS4.HLP
ADOBEPS5.DLL
ADOBEPSU.DLL
ADOBEPSU.HLP
DEFPRTR2.PPD
ICONLIB.DLL
PSMON.DLL
```

Copy these files to the */usr/share/cups/drivers* directory – you may need to rename some of the files so the filenames are all UPPERCASE.

Next, configure SAMBA (via the *smb.conf* file) to support printing through CUPS and provide a printer driver download share, as follows:

```
[global]
load printers = yes
printing = cups
printcap name = cups

[printers]
comment = All Printers
path = /var/spool/samba
browseable = no
public = yes
guest ok = yes
writable = no
printable = yes
printer admin = root
```

```
[print$]
comment = Printer Drivers
path = /etc/samba/drivers
browseable = yes
guest ok = no
read only = yes
write list = root
```

This configuration assumes a FHS−compliant installation of SAMBA; adjust the [printers] and [print$] share paths accordingly on your system as needed. That is, the directory for your printer drivers can be anywhere on the system; just make sure it is writable by the users specified by the `write list` directive plus readable and executable by all users. Also, make sure that you have SAMBA passwords defined for each user in the `write list` using SAMBA's `smbpasswd(1)` command. Otherwise you will not be able to authenticate.

Finally, run the `cupsaddsmb` command to export the printer drivers for one or more queues:

cupsaddsmb -U root printer1 ... printerN *ENTER*

Running `cupsaddsmb` with the −a option will export all printers:

cupsaddsmb -U root -a *ENTER*

Notice in the above examples that the user `root` was used which was defined in the `write list` of the *smb.conf* file.

Printing to Windows Servers

CUPS can print to Windows servers in one of two ways. The first way uses the LPD protocol on the CUPS system and the "TCP/IP Printing Services" on the Windows system. You can find out more about this configuration in the LPD section earlier in this chapter.

The second way is through the Microsoft Server Message Block ("SMB") protocol. Support for this protocol is provided with the free SAMBA software package. You can download SAMBA from:

http://www.samba.org

To configure CUPS for SAMBA, run the following command:

ln -s `which smbspool` /usr/lib/cups/backend/smb ENTER

The `smbspool(1)` program is provided with SAMBA starting with SAMBA 2.0.6. Once you have made the link you can configure your printers with one of the following device URIs:

```
smb://workgroup/server/sharename
smb://server/sharename
smb://user:pass@workgroup/server/sharename
smb://user:pass@server/sharename
```

The `workgroup` name need only be specified if your system is using a different workgroup. The `user:pass` strings are required when printing to Windows NT servers or to shares with passwords enabled under Windows 95 and 98.

A – Software License Agreement

Common UNIX Printing System License Agreement

Copyright 1997–2003 by Easy Software Products
44141 AIRPORT VIEW DR STE 204
HOLLYWOOD, MARYLAND 20636–3111 USA

Voice: +1.301.373.9600
Email: cups–info@cups.org
WWW: http://www.cups.org

Introduction

The Common UNIX Printing System^{TM}, ("CUPS^{TM}"), is provided under the GNU General Public License ("GPL") and GNU Library General Public License ("LGPL"), Version 2, with exceptions for Apple operating systems and the OpenSSL toolkit. A copy of the exceptions and licenses follow this introduction.

The GNU LGPL applies to the CUPS API library, located in the "cups" subdirectory of the CUPS source distribution and in the "cups" include directory and library files in the binary distributions. The GNU GPL applies to the remainder of the CUPS distribution, including the "pdftops" filter which is based upon Xpdf and the CUPS imaging library.

For those not familiar with the GNU GPL, the license basically allows you to:

- Use the CUPS software at no charge.
- Distribute verbatim copies of the software in source or binary form.
- Sell verbatim copies of the software for a media fee, or sell support for the software.
- Distribute or sell printer drivers and filters that use CUPS so long as source code is made available under the GPL.

What this license **does not** allow you to do is make changes or add features to CUPS and then sell a binary distribution without source code. You must provide source for any new drivers, changes, or additions to the software, and all code must be provided under the GPL or LGPL as appropriate. The only exceptions to this are the portions of the CUPS software covered by the Apple operating system license exceptions outlined later in this license agreement.

The GNU LGPL relaxes the "link−to" restriction, allowing you to develop applications that use the CUPS API library under other licenses and/or conditions as appropriate for your application.

License Exceptions

In addition, as the copyright holder of CUPS, Easy Software Products grants the following special exceptions:

1. **Apple Operating System Development License Exception**;
   a. Software that is developed by any person or entity for an Apple Operating System ("Apple OS−Developed Software"), including but not limited to Apple and third party printer drivers, filters, and backends for an Apple Operating System, that is linked to the CUPS imaging library or based on any sample filters or backends provided with CUPS shall not be considered to be a derivative work or collective work based on the CUPS program and is exempt from the mandatory source code release clauses of the GNU GPL. You may therefore distribute linked combinations of the CUPS imaging library with Apple OS−Developed Software without releasing the source code of the Apple OS−Developed Software. You may also use sample filters and backends provided with CUPS to develop Apple OS−Developed Software without releasing the source code of the Apple OS−Developed Software.
   b. An Apple Operating System means any operating system software developed and/or marketed by Apple Computer, Inc., including but not limited to all existing releases and versions of Apple's Darwin, Mac OS X, and Mac OS X Server products and all follow−on releases and future versions thereof.
   c. This exception is only available for Apple OS−Developed Software and does not apply to software that is distributed for use on other operating systems.
   d. All CUPS software that falls under this license exception have the following text at the top of each source file:

        This file is subject to the Apple OS−Developed Software exception.
2. **OpenSSL Toolkit License Exception**;
   a. Easy Software Products explicitly allows the compilation and distribution of the CUPS software with the OpenSSL Toolkit.

No developer is required to provide these exceptions in a derived work.

Trademarks

Easy Software Products has trademarked the Common UNIX Printing System, CUPS, and CUPS logo. These names and logos may be used freely in any direct port or binary distribution of CUPS. Please contract Easy

Software Products for written permission to use them in derivative products. Our intention is to protect the value of these trademarks and ensure that any derivative product meets the same high–quality standards as the original.

Binary Distribution Rights

Easy Software Products also sells rights to the CUPS source code under a binary distribution license for vendors that are unable to release source code for their drivers, additions, and modifications to CUPS under the GNU GPL and LGPL. For information please contact us at the address shown above.

The Common UNIX Printing System provides a "pdftops" filter that is based on the Xpdf software. For binary distribution licensing of this software, please contact:

> Derek B. Noonburg
> Email: derekn@foolabs.com
> WWW: http://www.foolabs.com/xpdf/

Support

Easy Software Products sells software support for CUPS as well as a commercial printing product based on CUPS called ESP Print Pro. You can find out more at our web site:

> `http://www.easysw.com/`

GNU GENERAL PUBLIC LICENSE

Version 2, June 1991

```
Copyright 1989, 1991 Free Software Foundation, Inc.
59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

Everyone is permitted to copy and distribute verbatim
copies of this license document, but changing it is not allowed.
```

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software––to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

GNU GENERAL PUBLIC LICENSE
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

9. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

   You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

8. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:
   a. You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.
   b. You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.
   c. if the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

   These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

   Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

   In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other

work under the scope of this License.

7. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

    a. Accompany it with the complete corresponding machine–readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

    b. Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine–readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

    c. Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

0. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

1. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

2. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

6. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty–free redistribution of the

10. Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

3. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

5. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

4. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR

REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

GNU LIBRARY GENERAL PUBLIC LICENSE

Version 2, June 1991

```
Copyright (C) 1991 Free Software Foundation, Inc.
59 Temple Place - Suite 330, Boston, MA  02111-1307, USA
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

[This is the first released version of the library GPL.  It is
 numbered 2 because it goes with version 2 of the ordinary GPL.]
```

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public Licenses are intended to guarantee your freedom to share and change free software−−to make sure the software is free for all its users.

This license, the Library General Public License, applies to some specially designated Free Software Foundation software, and to any other libraries whose authors decide to use it. You can use it for your libraries, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the library, or if you modify it.

For example, if you distribute copies of the library, whether gratis or for a fee, you must give the recipients all the rights that we gave you. You must make sure that they, too, receive or can get the source code. If you link a program with the library, you must provide complete object files to the recipients so that they can relink them with the library, after making changes to the library and recompiling it. And you must show them these terms so they know their rights.

Our method of protecting your rights has two steps: (1) copyright the library, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the library.

Also, for each distributor's protection, we want to make certain that everyone understands that there is no warranty for this free library. If the library is modified by someone else and passed on, we want its recipients to know that what they have is not the original version, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that companies distributing free software will individually obtain patent licenses, thus in effect transforming the program into proprietary software. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

Most GNU software, including some libraries, is covered by the ordinary GNU General Public License, which

was designed for utility programs. This license, the GNU Library General Public License, applies to certain designated libraries. This license is quite different from the ordinary one; be sure to read it in full, and don't assume that anything in it is the same as in the ordinary license.

The reason we have a separate public license for some libraries is that they blur the distinction we usually make between modifying or adding to a program and simply using it. Linking a program with a library, without changing the library, is in some sense simply using the library, and is analogous to running a utility program or application program. However, in a textual and legal sense, the linked executable is a combined work, a derivative of the original library, and the ordinary General Public License treats it as such.

Because of this blurred distinction, using the ordinary General Public License for libraries did not effectively promote software sharing, because most developers did not use the libraries. We concluded that weaker conditions might promote sharing better.

However, unrestricted linking of non–free programs would deprive the users of those programs of all benefit from the free status of the libraries themselves. This Library General Public License is intended to permit developers of non–free programs to use free libraries, while preserving your freedom as a user of such programs to change the free libraries that are incorporated in them. (We have not seen how to achieve this as regards changes in header files, but we have achieved it as regards changes in the actual functions of the Library.) The hope is that this will lead to faster development of free libraries.

The precise terms and conditions for copying, distribution and modification follow. Pay close attention to the difference between a "work based on the library" and a "work that uses the library". The former contains code derived from the library, while the latter only works together with the library.

Note that it is possible for a library to be covered by the ordinary General Public License rather than by this special one.

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License Agreement applies to any software library which contains a notice placed by the copyright holder or other authorized party saying it may be distributed under the terms of this Library General Public License (also called "this License"). Each licensee is addressed as "you".

A "library" means a collection of software functions and/or data prepared so as to be conveniently linked with application programs (which use some of those functions and data) to form executables.

The "Library", below, refers to any such software library or work which has been distributed under these terms. A "work based on the Library" means either the Library or any derivative work under copyright law: that is to say, a work containing the Library or a portion of it, either verbatim or with modifications and/or translated straightforwardly into another language. (Hereinafter, translation is included without limitation in the term "modification".)

"Source code" for a work means the preferred form of the work for making modifications to it. For a library, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the library.

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running a program using the Library is not restricted, and output from such a program is covered only if its contents constitute a work based on the Library (independent of the use of the Library in a tool for writing it). Whether that is true depends on what the Library does and what the program that uses the

Library does.

1. You may copy and distribute verbatim copies of the Library's complete source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and distribute a copy of this License along with the Library.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Library or any portion of it, thus forming a work based on the Library, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

> d. The modified work must itself be a software library.

> c. You must cause the files modified to carry prominent notices stating that you changed the files and the date of any change.

> b. You must cause the whole of the work to be licensed at no charge to all third parties under the terms of this License.

> a. If a facility in the modified Library refers to a function or a table of data to be supplied by an application program that uses the facility, other than as an argument passed when the facility is invoked, then you must make a good faith effort to ensure that, in the event an application does not supply such function or table, the facility still operates, and performs whatever part of its purpose remains meaningful.

> (For example, a function in a library to compute square roots has a purpose that is entirely well–defined independent of the application. Therefore, Subsection 2d requires that any application–supplied function or table used by this function must be optional: if the application does not supply it, the square root function must still compute square roots.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Library, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Library, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Library.

In addition, mere aggregation of another work not based on the Library with the Library (or with a work based on the Library) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may opt to apply the terms of the ordinary GNU General Public License instead of this License to a given copy of the Library. To do this, you must alter all the notices that refer to this License, so that they refer to the ordinary GNU General Public License, version 2, instead of to this License. (If a newer version than

version 2 of the ordinary GNU General Public License has appeared, then you can specify that version instead if you wish.) Do not make any other change in these notices.

Once this change is made in a given copy, it is irreversible for that copy, so the ordinary GNU General Public License applies to all subsequent copies and derivative works made from that copy.

This option is useful when you wish to copy part of the code of the Library into a program that is not a library.

4. You may copy and distribute the Library (or a portion or derivative of it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you accompany it with the complete corresponding machine−readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange.

If distribution of object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place satisfies the requirement to distribute the source code, even though third parties are not compelled to copy the source along with the object code.

5. A program that contains no derivative of any portion of the Library, but is designed to work with the Library by being compiled or linked with it, is called a "work that uses the Library". Such a work, in isolation, is not a derivative work of the Library, and therefore falls outside the scope of this License.

However, linking a "work that uses the Library" with the Library creates an executable that is a derivative of the Library (because it contains portions of the Library), rather than a "work that uses the library". The executable is therefore covered by this License. Section 6 states terms for distribution of such executables.

When a "work that uses the Library" uses material from a header file that is part of the Library, the object code for the work may be a derivative work of the Library even though the source code is not. Whether this is true is especially significant if the work can be linked without the Library, or if the work is itself a library. The threshold for this to be true is not precisely defined by law.

If such an object file uses only numerical parameters, data structure layouts and accessors, and small macros and small inline functions (ten lines or less in length), then the use of the object file is unrestricted, regardless of whether it is legally a derivative work. (Executables containing this object code plus portions of the Library will still fall under Section 6.)

Otherwise, if the work is a derivative of the Library, you may distribute the object code for the work under the terms of Section 6. Any executables containing that work also fall under Section 6, whether or not they are linked directly with the Library itself.

6. As an exception to the Sections above, you may also compile or link a "work that uses the Library" with the Library to produce a work containing portions of the Library, and distribute that work under terms of your choice, provided that the terms permit modification of the work for the customer's own use and reverse engineering for debugging such modifications.

You must give prominent notice with each copy of the work that the Library is used in it and that the Library and its use are covered by this License. You must supply a copy of this License. If the work during execution displays copyright notices, you must include the copyright notice for the Library among them, as well as a reference directing the user to the copy of this License. Also, you must do one of these things:

    d. Accompany the work with the complete corresponding machine−readable source code for the Library including whatever changes were used in the work (which must be distributed under Sections 1 and 2

above); and, if the work is an executable linked with the Library, with the complete machine−readable "work that uses the Library", as object code and/or source code, so that the user can modify the Library and then relink to produce a modified executable containing the modified Library. (It is understood that the user who changes the contents of definitions files in the Library will not necessarily be able to recompile the application to use the modified definitions.)

    c. Accompany the work with a written offer, valid for at least three years, to give the same user the materials specified in Subsection 6a, above, for a charge no more than the cost of performing this distribution.

    b. If distribution of the work is made by offering access to copy from a designated place, offer equivalent access to copy the above specified materials from the same place.

    a. Verify that the user has already received a copy of these materials or that you have already sent this user a copy.

For an executable, the required form of the "work that uses the Library" must include any data and utility programs needed for reproducing the executable from it. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

It may happen that this requirement contradicts the license restrictions of other proprietary libraries that do not normally accompany the operating system. Such a contradiction means you cannot use both them and the Library together in an executable that you distribute.

7. You may place library facilities that are a work based on the Library side−by−side in a single library together with other library facilities not covered by this License, and distribute such a combined library, provided that the separate distribution of the work based on the Library and of the other library facilities is otherwise permitted, and provided that you do these two things:

    b. Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities. This must be distributed under the terms of the Sections above.

    a. Give prominent notice with the combined library of the fact that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

8. You may not copy, modify, sublicense, link with, or distribute the Library except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, link with, or distribute the Library is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

9. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Library or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Library (or any work based on the Library), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Library or works based on it.

10. Each time you redistribute the Library (or any work based on the Library), the recipient automatically receives a license from the original licensor to copy, distribute, link with or modify the Library subject to

these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

11. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Library at all. For example, if a patent license would not permit royalty–free redistribution of the Library by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Library.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply, and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

12. If the distribution and/or use of the Library is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Library under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

13. The Free Software Foundation may publish revised and/or new versions of the Library General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Library does not specify a license version number, you may choose any version ever published by the Free Software Foundation.

14. If you wish to incorporate parts of the Library into other free programs whose distribution conditions are incompatible with these, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

15. BECAUSE THE LIBRARY IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE LIBRARY, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN

OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE LIBRARY "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE LIBRARY IS WITH YOU. SHOULD THE LIBRARY PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE LIBRARY AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE LIBRARY (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE LIBRARY TO OPERATE WITH ANY OTHER SOFTWARE), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

GNU LIBRARY GENERAL PUBLIC LICENSE

B – Common Network Settings

This appendix covers many of the popular TCP/IP network interfaces and printer servers available on the market today.

Configuring a Network Interface

When you first install a network printer or print server on your LAN, you need to set the Internet Protocol ("IP") address. On most higher–end "workgroup" printers, you can set the address through the printer control panel. However, in most cases you will want to assign the addresses remotely from your workstation. This makes administration a bit easier and avoids assigning duplicate addresses accidentally.

To setup your printer or print server for remote address assignment, you'll need the Ethernet Media Access Control ("MAC") address, also sometimes called a node address, and the IP address you want to use for the device. The Ethernet MAC address can often be found on the printer test page or bottom of the print server.

Configuring the IP Address Using ARP

The easiest way to set the IP address of a network device is to use the `arp(8)` command. The `arp` sends an Address Resolution Protocol ("ARP") packet to the specified Ethernet MAC address, setting the network device's IP address:

```
arp -s ip-address ethernet-address ENTER
arp -s host.domain.com 08:00:69:00:12:34 ENTER
arp -s 192.0.2.2 08:00:69:00:12:34 ENTER
```

Configuring the IP Address Using RARP

The most flexible way to remotely assign IP addresses under UNIX is through the Reverse Address Resolution Protocol ("RARP"). RARP allows a network device to request an IP address using its Ethernet MAC address, and one or more RARP servers on the network will respond with an ARP packet with the IP address the device can use.

RARP should be used when you have to manage many printers or print servers, or when you have a network device that does not remember its IP address after a power cycle. If you just have a single printer or print server, the `arp` command is the way to go.

Some UNIX operating systems use a program called `rarpd(8)` to manage RARP. Others, like Linux, support this protocol in the kernel. For systems that provide the `rarpd` program you will need to start it before RARP lookups will work:

```
rarpd ENTER
```

Under IRIX you can enable this functionality by default using:

```
chkconfig rarpd on ENTER
```

Both the `rarpd` program and kernel RARP support read a list of Ethernet and IP addresses from the file */etc/ethers*. Each line contains the Ethernet address (colon delimited) followed by an IP address or hostname like:

```
08:00:69:00:12:34 myprinter.mydomain.com
08:00:69:00:12:34 192.0.2.2
```

Add a line to this file and cycle the power on the printer or print server to set its address.

Configuring the IP Address Using BOOTP

The BOOTP protocol is used when you need to provide additional information such as the location of a configuration file to the network interface. Using the standard `bootpd(8)` program supplied with UNIX you simply need to add a line to the */etc/bootptab* file; for IRIX:

```
myprinter 08:00:69:00:12:34 192.0.2.2 myprinter.boot
```

Newer versions of `bootpd` use a different format:

```
myprinter:ha=080069001234:ip=192.0.2.2:t144=myprinter.boot
```

The *myprinter.boot* file resides in the */usr/local/boot* directory by default. If you do not need to provide a boot file you may leave the last part of the line blank.

> **NOTE:**
>
> Some versions of UNIX do not enable the BOOTP service by default. The */etc/inetd.conf* usually contains a line for the BOOTP service that can be uncommented if needed.

Verifying the Printer Connection

To test that the IP address has been successfully assigned and that the printer is properly connected to your LAN, type:

```
ping ip-address ENTER
```

If the connection is working properly you will see something like:

```
ping myprinter ENTER
PING myprinter (192.0.2.2): 56 data bytes
64 bytes from 192.0.2.2: icmp_seq=0 ttl=15 time=5 ms
64 bytes from 192.0.2.2: icmp_seq=1 ttl=15 time=3 ms
64 bytes from 192.0.2.2: icmp_seq=2 ttl=15 time=3 ms
64 bytes from 192.0.2.2: icmp_seq=3 ttl=15 time=3 ms
```

If not, verify that the printer or print server is connected to the LAN, it is powered on, the LAN cabling is good, and the IP address is set correctly. You can usually see the current IP address and network status by printing a configuration or test page on the device.

Common Network Interface Settings

Once you have set the IP address you can access the printer or print server using the `ipp`, `lpd`, or `socket` backends. The following is a list of common network interfaces and printer servers and the settings you should use with CUPS:

Model/Manufacturer	Device URI(s)
Apple LaserWriter	lpd://*address*/PASSTHRU
Axis w/o IPP (see directions)	socket://*address*:9100 socket://*address*:9101 socket://*address*:9102
Axis w/IPP	ipp://*address*/LPT1 ipp://*address*/LPT2 ipp://*address*/COM1
Castelle LANpress^{TM}	lpd://*address*/pr1 lpd://*address*/pr2 lpd://*address*/pr3
DPI NETPrint	lpd://*address*/pr1 lpd://*address*/pr2 lpd://*address*/pr3
EFI® Fiery® RIP	lpd://*address*/print
EPSON® Multiprotocol Ethernet Interface Board	socket://*address*
Extended System ExtendNET	lpd://*address*/pr1 lpd://*address*/pr2 lpd://*address*/pr3
Hewlett Packard JetDirect w/o IPP	socket://*address*:9100 socket://*address*:9101 socket://*address*:9102
Hewlett Packard JetDirect w/IPP	ipp://*address*/ipp ipp://*address*/ipp/port1 ipp://*address*/ipp/port2 ipp://*address*/ipp/port3
Intel® NetportExpress XL, PRO/100	lpd://*address*/LPT1_PASSTHRU lpd://*address*/LPT2_PASSTHRU lpd://*address*/COM1_PASSTHRU
Lexmark^{TM} MarkNet	lpd://*address*/ps
Linksys EtherFast® (see directions)	socket://*address*:4010 socket://*address*:4020 socket://*address*:4030
Kodak®	lpd://*address*/ps
QMS® CrownNet^{TM}	lpd://*address*/ps
Tektronix® PhaserShare^{TM}	socket://*address*:9100
XEROX® 4512 NIC	lpd://*address*/PORT1
XEROX® XNIC	lpd://*address*/PASSTHRU
XEROX® (most others)	socket://*address*:5503

Configuring Axis Print Servers

The Axis print servers can be configured using ARP, RARP, or BOOTP. However, on models that do not provide IPP support an additional step must be performed to configure the TCP/IP portion of the print server for use with CUPS.

Each print server contains a configuration file named *config* that contains a list of network parameters used by the server. To modify this file you must first download it from the print server using the `ftp(1)` program:

```
ftp ip-address ENTER
Connected to ip-address.
220 Axis NPS ### FTP Printer Server V#.## MON DD YEAR ready.
ftp> user root ENTER
331 User name ok, need password
Password: pass ENTER (this is not echoed)
230 User logged in
ftp> get config ENTER
local: config remote: config
200 PORT command successful.
150 Opening data connection for config (192,0,2,2),
(mode ascii).
226 Transfer complete.
##### bytes received in #.## seconds (##### Kbytes/s)
ftp> quit ENTER
221 Goodbye.
```

Next, edit the file with your favorite text editor and locate the lines beginning with:

```
RTN_OPT.       : YES
RTEL_PR1.      : 0
RTEL_PR2.      : 0
RTEL_PR3.      : 0
RTEL_PR4.      : 0
RTEL_PR5.      : 0
RTEL_PR6.      : 0
RTEL_PR7.      : 0
RTEL_PR8.      : 0
```

Change the `RTN_OPT` line to read:

```
RTN_OPT.       : NO
```

This disables the Reverse TELNET protocol and enables the standard TELNET protocol on the print server. Next, assign a port number for each parallel and serial port on the server as follows:

```
RTEL_PR1.      : 9100
RTEL_PR2.      : 9101
RTEL_PR3.      : 9102
RTEL_PR4.      : 9103
RTEL_PR5.      : 9104
RTEL_PR6.      : 9105
RTEL_PR7.      : 9106
RTEL_PR8.      : 9107
```

This essentially makes the Axis print server look like a Hewlett Packard JetDirect EX print server. Save the file and then upload the new *config* file using the `ftp` command:

```
ftp ip-address ENTER
Connected to ip-address.
220 Axis NPS ### FTP Printer Server V#.## MON DD YEAR ready.
ftp> user root ENTER
331 User name ok, need password
Password: pass ENTER (this is not echoed)
230 User logged in
ftp> put config CONFIG ENTER
local: config remote: CONFIG
200 PORT command successful.
150 Opening data connection for config (192,0,2,2), (mode ascii).
226 Transfer complete.
##### bytes received in #.## seconds (##### Kbytes/s)
ftp> get hardreset ENTER
local: hardreset remote: hardreset
200 PORT command successful.
421 Axis NPS ### hard reset, closing connection.
ftp> quit ENTER
221 Goodbye.
```

Your Axis print server is now ready for use!

Configuring Linksys Print Servers

The Linksys print servers can be configured using ARP, RARP, or BOOTP. Like older Axis print servers, an additional step must be performed to configure the TCP/IP portion of the print server for use with CUPS.

Each print server contains a configuration file named *CONFIG* that contains a list of network parameters used by the server. To modify this file you must first download it from the print server using the `ftp(1)` program:

```
ftp -n ip-address ENTER
Connected to ip-address.
220 Print Server Ready.
Remote system type is Print.
ftp> get CONFIG ENTER
local: CONFIG remote: CONFIG
200 Command OK.
150 Open ASCII Mode Connection.
WARNING! 68 bare linefeeds received in ASCII mode
File may not have transferred correctly.
226 Transfer complete.
##### bytes received in #.## seconds (##### Kbytes/s)
ftp> quit ENTER
221 Goodbye.
```

Next, edit the file with your favorite text editor and locate the lines beginning with:

```
0100 L1_PROUT:P1
0120 L2_PROUT:P1
0140 L3_PROUT:P1
```

Change the port number for each parallel and serial port on the server as follows:

```
0100 L1_PROUT:P1
0120 L2_PROUT:P2
0140 L3_PROUT:P3
```

This maps each virtual printer with a physical port. Save the file and then upload the new *CONFIG* file using the `ftp` command:

```
ftp -n ip-address ENTER
Connected to ip-address.
220 Print Server Ready.
Remote system type is Print.
ftp> put CONFIG ENTER
local: CONFIG remote: CONFIG
200 Command OK.
150 Open ASCII Mode Connection.
226 Transfer complete.
##### bytes received in #.## seconds (##### Kbytes/s)
ftp> quit ENTER
221 Goodbye.
```

Your Linksys print server is now ready for use!

Configuring Linksys Print Servers

C – Printer Drivers

This appendix lists the printer drivers that are provided with CUPS.

Printer Drivers

CUPS includes the following printer drivers:

- <u>EPSON 9–pin Dot Matrix</u>, *epson9.ppd*
- <u>EPSON 24–pin Dot Matrix</u>, *epson24.ppd*
- <u>EPSON Stylus Color</u>, *stcolor.ppd*
- <u>EPSON Stylus Photo</u>, *stphoto.ppd*
- <u>HP DeskJet</u>, *deskjet.ppd*
- <u>HP LaserJet</u>, *laserjet.ppd*

EPSON 9–pin Dot Matrix

The EPSON 9–pin Dot Matrix driver (*epson9.ppd*) supports 9–pin dot matrix printers that implement the ESC/P command set. It provides 60x72, 120x72, and 240x72 DPI output in black only.

EPSON 24–pin Dot Matrix

The EPSON 24–pin Dot Matrix driver (*epson9.ppd*) supports 24–pin dot matrix printers that implement the ESC/P command set. It provides 120x180, 180x180, 360x180, and 360x360 DPI output in black only.

EPSON Stylus Color

The EPSON Stylus Color driver (*stcolor.ppd*) supports EPSON Stylus Color printers that implement the ESC/P2 command set. It provides 180, 360, and 720 DPI output in black and color (CMYK).

EPSON Stylus Photo

The EPSON Stylus Photo driver (*stphoto.ppd*) supports EPSON Stylus Photo printers that implement the ESC/P2 command set. It provides 180, 360, and 720 DPI output in black and color (CMYKcm).

HP DeskJet

The HP DeskJet driver (*deskjet.ppd*) supports HP DeskJet printers that implement the PCL command set. It provides 150, 300, and 600 DPI output in black and color (CMYK).

The DeskJet printers that implement the HP−PPA command set (720C, 722C, 820C, and 1100C) are **not** supported due to a complete lack of documentation and support from Hewlett Packard.

The duplexer provided with the HP DeskJet 900 series printers is also not supported for similar reasons.

HP LaserJet

The HP LaserJet driver (*laserjet.ppd*) supports HP LaserJet printers that implement the PCL command set. It provides 150, 300, and 600 DPI output in black only and supports the duplexer if installed.

LaserJet printers that do not implement PCL (3100, 3150) are not supported due to a complete lack of documentation and support from Hewlett Packard.

D – List of Files

This appendix lists the files and directories that are installed for the Common UNIX Printing System.

Pathname	Description
/etc/cups/certs/	The location of authentication certificate files for local HTTP clients.
/etc/cups/classes.conf	The printer classes configuration file for the scheduler.
/etc/cups/cupsd.conf	The scheduler configuration file.
/etc/cups/interfaces/	The location of System V interface scripts for printers.
/etc/cups/mime.convs	The list of standard file filters included with CUPS.
/etc/cups/mime.types	The list of recognized file types for CUPS.
/etc/cups/ppd/	The location of PostScript Printer Description ("PPD") files for printers.
/etc/cups/printers.conf	The printer configuration file for the scheduler.
/usr/bin/cancel	The System V cancel job(s) command.
/usr/bin/disable	The System V disable printer command.
/usr/bin/enable	The System V enable printer command.

/usr/bin/lp	The System V print command.
/usr/bin/lpoptions	Sets user−defined printing options and defaults.
/usr/bin/lppasswd	Adds, changes, or removes Digest password accounts.
/usr/bin/lpq	The Berkeley status command.
/usr/bin/lpr	The Berkeley print command.
/usr/bin/lprm	The Berkeley cancel job(s) command.
/usr/bin/lpstat	The System V status command.
/usr/include/cups/	CUPS API header files.
/usr/lib32/libcups.a /usr/lib32/libcupsimage.a	Static libraries (IRIX 6.5)
/usr/lib/libcups.a /usr/lib/libcupsimage.a	Static libraries (all others)
/usr/lib/libcups.sl.2 /usr/lib/libcupsimage.sl.2	Shared libraries (HP−UX)
/usr/lib32/libcups.so.2 /usr/lib32/libcupsimage.so.2	Shared libraries (IRIX 6.5)
/usr/lib/libcups.so.2 /usr/lib/libcupsimage.so.2	Shared libraries (all others)
/usr/lib/cups/backend/	Backends for various types of printer connections.
/usr/lib/cups/cgi−bin/	CGI programs for the scheduler.
/usr/lib/cups/daemon/	Daemons for polling and LPD support.
/usr/lib/cups/filter/	Filters for various types of files.
/usr/lib/locale/	The location of language−specific message files. (System V)
/usr/lib/nls/msg/	The location of language−specific message files. (Compaq Tru64 UNIX)
/usr/share/locale/	The location of language−specific message files. (Linux, *BSD)
/usr/sbin/accept	The accept−jobs command.
/usr/sbin/cupsd	The CUPS print scheduler.
/usr/sbin/lpadmin	The System V printer administration tool.
/usr/sbin/lpc	The Berkeley printer administration tool.
/usr/sbin/lpinfo	The get−devices and get−ppds command.
/usr/sbin/lpmove	The move−jobs command.
/usr/sbin/reject	The reject−jobs command.
/usr/share/catman/a_man/ /usr/share/catman/u_man/	Man pages (IRIX)
/usr/share/man/	Man pages (Compaq Tru64 UNIX, HP−UX, Solaris)
/usr/man/	Man pages (all others)
/usr/share/cups/data/	The location of filter data files.

/usr/share/cups/data/testprint.ps	The PostScript test page file.
/usr/share/cups/fonts/	The location of PostScript fonts for the PostScript RIP.
/usr/share/cups/model/	The location of PostScript Printer Description ("PPD") files and interface scripts that may be used to setup a printer queue.
/usr/share/cups/pstoraster/	Other PostScript RIP initialization files.
/usr/share/cups/pstoraster/Fontmap	The font mapping file (converts filenames to fontnames)
/usr/share/cups/templates/	The location of HTML template files for the web interfaces.
/usr/share/doc/cups/	Documentation and web page data for the scheduler.
/var/log/cups/	The location of scheduler log files.
/var/spool/cups/	The location of print files waiting to be printed.

E – Troubleshooting Common Problems

This appendix covers some of the common problems first–time users encounter when installing and configuring CUPS.

Commercial support for CUPS is available from Easy Software Products. For more information please contact us at:

- WWW: http://www.easysw.com
- EMail: info@easysw.com
- Telephone (M–F, 9–5 EST): +1.301.373.9600

My Applications Don't See the Available Printers

Many applications read the */etc/printcap* file to get a list of available printers.

The default CUPS configuration creates the */etc/printcap* file automatically. To enable or disable automatic creation and updating of this file, use the Printcap directive described in Chapter 6, "Printing System Management".

CUPS Doesn't Recognize My Username or Password!

CUPS will ask you for a UNIX username and password when you perform printer administration tasks remotely or via a web browser. The default configuration requires that you use the root username and the corresponding password to authenticate the request.

CUPS does not allow you to authenticate an administration request with an account that has no password for security reasons. If you do not have a password on your `root` account then you won't be able to add printers remotely or via the web interface!

To disable password authentication you need to edit the */etc/cups/cupsd.conf* file and comment out the lines reading:

```
AuthType Basic
AuthClass System
```

for the */admin* location. Then restart the CUPS server as described in Chapter 6, "Printing System Management".

> **NOTE:**
>
> Disabling password checks will allow any local user to change your printer and class configuration, but remote administration from another machine will still not be allowed.

I Can't Do Administration Tasks from Another Machine!

The default CUPS configuration limits administration to the local machine. To open up access, edit the */etc/cups/cupsd.conf* and comment out the lines reading:

```
Order deny,allow
Deny from all
Allow from 127.0.0.1
```

for the */admin* location. Then restart the CUPS server as described in Chapter 6, "Printing System Management".

> **NOTE:**
>
> Allowing administration access from all hosts is a potential security risk. Please read Chapter 6, "Printing System Management" for a description of these risks and ways to minimize them.

I Can't Do Administration Tasks from My Web Browser!

This problem is usually caused by:

1. not specifying the correct password for the root account.
2. accessing the CUPS server using the hostname or IP address of the server without enabling remote access for administration functions. This can be corrected by following the instructions in the "I Can't Do Administration Tasks from Another Machine!" section earlier in this appendix.
3. not setting a password on the root account. CUPS will not authenticate a user account that does not have a password for security reasons.
4. authenticating using an account other than root, but the account you are using is not a member of the system group.
5. configuring CUPS to use Digest authentication, but your web browser does not support Digest authentication.

Connection Refused Messages

Under normal circumstances, "connection refused" messages for a networked printer should be expected from time to time. Most network interfaces only allow a single connection to be made at any given time (one job at a time) and will refuse access to all other systems while the first connection is active. CUPS automatically retries the connection once every 30 seconds.

If the problem persists and you are unable to print any jobs to the printer, verify that another machine is not maintaining a connection with the printer, and that you have selected the proper port or printer name for the printer.

Also, most external print servers will refuse connections if the connected printer is turned off or is off–line. Verify that the affected printer is turned on and is online.

Write Error Messages

If you get "write error" messages on a printer queue the printer interface (usually a Hewlett Packard JetDirect interface) has timed out and reset the network connection from your workstation.

The error is caused by that startup delay between the initial setup of the printer or plotter and the first page of print data that is sent.

To correct the problem, change the idle timeout on the interface to at least 180 seconds or 3 minutes. To change the timeout on a Hewlett Packard JetDirect interface, type:

```
telnet ip-address ENTER

Trying ip-address...
Connected to ip-address.
Escape character is `^]'.

Please type [Return] two times, to initialize telnet configuration
For HELP type "?"
> idle-timeout: 180 ENTER
> quit ENTER
```