

**FORSCHUNGSZENTRUM JÜLICH GmbH**  
**Zentralinstitut für Angewandte Mathematik**  
**D-52425 Jülich, Tel. (02461) 61-6402**

Interner Bericht

**Einrichten eines Linux Printservers  
mit dem  
Common Unix Printing System**

*Stephan Graf*

FZJ-ZAM-IB-2007-01

Januar 2007

(letzte Änderung: 24.1.2007)



# Inhaltsverzeichnis

<b>1</b>	<b>Internet Printing Protokoll</b>	<b>3</b>
1.1	Historie . . . . .	3
1.2	Anforderungen an das neue Protokoll . . . . .	3
1.3	Realisierung des IPP . . . . .	4
<b>2</b>	<b>Das Common Unix Printing System</b>	<b>5</b>
2.1	Konzept . . . . .	5
2.2	Installation von CUPS 1.1.23 auf einem SuSE Linux Server . . . . .	6
2.3	Installation und Konfiguration eines neuen Druckers . . . . .	7
2.3.1	Web-Interface . . . . .	7
2.3.2	Installation mit System-Kommandos . . . . .	9
2.4	CUPS-Konfigurationsdateien . . . . .	11
2.4.1	/etc/cups/cupsd.conf . . . . .	11
2.4.2	/etc/cups/client.conf . . . . .	13
2.4.3	/etc/cups/printers.conf . . . . .	13
2.4.4	/etc/cups/lpoptions . . . . .	14
2.4.5	/etc/cups/classes.conf . . . . .	14
2.4.6	/etc/cups/mime.types und /etc/cups/mime.convs . . . . .	15
2.5	CUPS-Backend – Kommunikation mit den Druckern . . . . .	15
2.5.1	Erstellen eigener Backends . . . . .	16
2.6	Bannerpages in CUPS . . . . .	16
2.6.1	Bannerpages selbst erstellen . . . . .	16
<b>3</b>	<b>Microsoft Windows und CUPS</b>	<b>19</b>
3.1	Drucken via LPR/LPD . . . . .	19
3.2	Drucken via IPP . . . . .	20
3.3	CUPS und Samba . . . . .	21
3.3.1	Konfigurationsdatei <i>smb.conf</i> . . . . .	21
3.3.2	Installation unter Windows . . . . .	23
3.4	Alternative: CupsClient . . . . .	24
<b>4</b>	<b>Tipps rund um CUPS</b>	<b>27</b>
4.1	FTP-Backend für CUPS . . . . .	27
4.2	variables Backend – <i>Tea4Cups</i> . . . . .	29
4.2.1	Installation von <i>Tea4Cups</i> . . . . .	29
4.2.2	Konfiguration von <i>Tea4Cups</i> . . . . .	30
4.3	Tipps zur Erstellung eines Deckblattes . . . . .	31



# Abbildungsverzeichnis

1	Konzept der ZAM-Printserver basierend auf LPR . . . . .	1
2.1	CUPS – Ablaufdiagramm . . . . .	5
2.2	Jobverlauf . . . . .	6
2.3	Web-Interface des CUPS-Dienstes . . . . .	7
2.4	Druckerinstallation via Web-Interface . . . . .	8
2.5	Status des neu installierten Druckers . . . . .	8
2.6	Konfiguration des Druckers . . . . .	9
2.7	lpoptions -p zam09 -l . . . . .	11
2.8	Einträge zum Drucker „zam09“ in „printers.conf“ . . . . .	13
2.9	Beispiel-Einträge in der „lpoptions“-Datei . . . . .	14
2.10	Klasse „zam“ in „classes.conf“ . . . . .	14
2.11	PDF Dateien beginnen mit PDF (Zeile aus „mimes.types“) . . . . .	15
2.12	pdftops wandelt PDF in PostScript um (Zeile aus „mimes.convs“) . . . . .	15
2.13	Fachauswahl für eine <i>HP 8500</i> in einer Bannerpaper . . . . .	17
3.1	Unix Druckdienste für Windows . . . . .	19
3.2	Windows LPR-Drucker einrichten . . . . .	20
3.3	IPP-Drucker unter Windows installieren . . . . .	20
3.4	CUPS – Ablaufdiagramm mit Samba-Server . . . . .	21
3.5	„global“-Sektion der „smb.conf“ Datei . . . . .	22
3.6	Abschnitt für das Drucker-Share <i>zam09</i> . . . . .	22
3.7	Abschnitt für die Druckertreiber . . . . .	23
3.8	Samba Drucker unter Windows einrichten . . . . .	24
3.9	Drucker installieren mit <i>CupsClient</i> . . . . .	24
4.1	Beispiel für die Datei <i>tea4cups.conf</i> . . . . .	30



# Tabellenverzeichnis

2.1	Optionen des <i>lpadmin</i> -Befehls . . . . .	10
2.2	Optionen des <i>lpoptions</i> -Befehls . . . . .	10
2.3	Auswahl an Parametern der „ <i>cupsd.conf</i> “-Datei . . . . .	12
2.4	Backends von CUPS . . . . .	15
4.1	Variablen von Tea4Cups . . . . .	30



# Vorwort

Das Printing-Geschäft ist ein undankbares Feld. Das Drucken muß funktionieren! Keiner will sich damit lange aufhalten, schließlich funktioniert es Zuhause ja auch immer mittels Plug&Play.

Aber in einem heterogenen Umfeld mit vielen verschiedenen Betriebssystemen, Anwendungen und auch vielen verschiedenen Druckermodellen sieht die Welt schon anders aus. Das alles unter einen Hut zu bringen ist nicht trivial und fordert einigen Aufwand und Know-How.

Auf unterschiedliche Weisen können Drucker von einem Rechner aus genutzt werden. So werden auch im Forschungszentrum Jülich Drucker lokal eingesetzt, z.B. wenn sensible Daten gedruckt werden sollen. Daneben gibt es aber die Abteilungs- und Institutsdrucker, und für professionelle Ausgaben wie Poster oder kleinere Auflagen von Büchern die zentralen Drucker der Graphischen Betriebe (B-SG). Diese werden über das Netzwerk angesprochen.

Das Zentralinstitut für Angewandte Mathematik (ZAM) bietet den anderen Instituten des Forschungszentrum an, deren Drucker über einen zentralen Server anzusteuern. Der Vorteil ist, daß sich der Benutzer/Administrator nicht mehr um die Anbindung der ganzen Drucker kümmern muß, sondern anhand einer Anleitung ( $\rightarrow$ [L4]) jeden verfügbaren Drucker installieren kann. Jedes Institut kann selbst entscheiden, ob es den Service des ZAM nutzen oder selber einen Printserver installieren möchte.

Auf den zentralen Printservern läuft eine Mitte der 80er Jahre im ZAM entwickelte Software basierend auf dem LPR-Protokoll, das in der UNIX-Welt schon um 1975 existierte. Der Server stellt die Drucker der Welt zur Verfügung, nimmt die Jobs an, bereitet diese auf und schickt sie zum Zieldrucker.

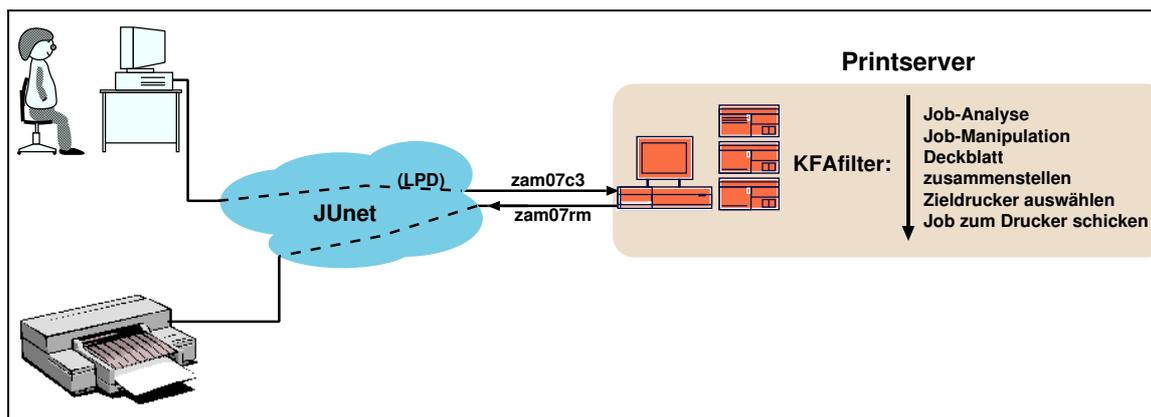


Abbildung 1: Konzept der ZAM-Printserver basierend auf LPR

Doch die Entwicklungen sind weitergegangen. Die Drucker besitzen heute wesentlich mehr Fähigkeiten und es ist relativ aufwendig, diese über ein LPR-basierenden Printing-Dienst anzusprechen. Deswegen wurde ein neuer Standard entwickelt, der die modernen Anforderungen an das Drucken erfüllt: Das *Internet Printing Protocol*

Darauf basiert ein frei verfügbares Softwarepaket – das Common Unix Printing System – welches auf verschiedenen UNIX-Plattformen und natürlich auch auf Linux installiert werden kann.

Im folgenden wird beschrieben, was bei einer Installation zu beachten ist, um einen Printserver zu

konfigurieren, der den Anforderungen eines Forschungszentrum gewachsen ist.

# Kapitel 1

## Internet Printing Protokoll

### 1.1 Historie

Da die ersten Drucker einfache Zeilendrucker waren, war die Anforderung an das Printing nicht sonderlich hoch. Ab etwa 1975 wurde in den verschiedenen UNIX-Derivaten ein Printing-Protokoll realisiert, daß das Drucken via Netz erlaubte: *Line Printer Daemon (LPD) / Line Printer Remote (LPR)*

Vom Grundkonzept gleich, entwickelten die zwei großen Lager (IBM: System-V, Berkley University: BSD) leicht unterschiedliche Varianten.

Erst 1990 wurde das Protokoll als RFC 1179 (LPD) zum offiziellen Standard erhoben, blieb aber bis heute in dem Zustand „informativ“

Doch die EDV-Landschaft entwickelte sich weiter und heute sind die Anforderungen an den Zugriff auf Ausgabegeräte gestiegen.

So bieten Drucker eine Vielzahl an Optionen, sei es Duplex-Printing, verschiedene Medien, unterschiedliche Qualität, automatisches Heften, usw. Dazu kommt die Kommunikation. Gedruckt wird via Intranet, das den Mitarbeitern in der Regel zur Verfügung steht. Also müssen sensible Daten verschlüsselt werden. Oder es sollen nur privilegierte Benutzer einen speziellen Drucker nutzen können. Diese Features sind beim klassischen LPD nicht vorgesehen. Zwar gibt es eine Reihe an Weiterentwicklungen wie *LPRng*, *PPR*, *PLP*, *GNUlpr* und *PDQ*, doch basieren die alle auf dem alten Standard RFC 1179, dessen Schwächen sie zu beheben versuchen.

Dieses Problem wurde in den 90ern erkannt und verschiedene Firmen – u.a. Sharp, HP, Xerox und Microsoft – haben sich zur *Printer Working Group* zusammengetan und ein neues Protokoll entwickelt.

Im April 1999 wurde das Internet Printing Protocoll (IPP) von der *Internet Engineering Task Force (IETF)* zum offiziellen Standard erhoben (RFC 2565). Die Version 1.0 war noch im Zustand „experimental“, die Version 1.1 besitzt inzwischen in dem Status „proposed standard“ (RFC 2910).[L5, L6]

### 1.2 Anforderungen an das neue Protokoll

Voraussetzung für einen neuen Standard war, das er abwärts kompatibel ist. D.h. in erster Linie, daß alle alten Features des LPD/LPR-Protokolls ebenfalls realisiert sein müssen. Dazu zählen:

- Druckjobs an den ausgewählten Drucker schicken (die sogenannte Push-Methode)
- einen zuvor geschickten Job wieder annullieren

- moderne PDLs<sup>1</sup> (PostScript, PCL,...) verarbeiten
- auf Anforderung interne Anweisungen der PDLs mit eigenen überschreiben
- Rückmeldungen des Druckers bei Problemen entgegennehmen
- den Fortgang eines Jobs überwachen

Darüber hinaus wurden folgende neuen Anforderungen an das IPP gestellt:

- einen IPP-Drucker oder -Server anweisen, das zu druckende Dokument selbst abzuholen (Pull-Methode)
- Printer im Netzwerk finden, selbst wenn sie vorher noch nicht bekannt sind
- eine Liste der besonderen Fähigkeiten eines bestimmten Druckers abrufen
- Informationen über Treiber für den ausgewählten Drucker verfügbar machen
- das Drucken bereits beginnen, noch bevor alle Dokumentendaten übertragen sind
- mehrere Dokumente zu einem einzigen Druckjob zusammenbinden
- den Fortgang eines Jobs von überall her überwachen
- Drucker und Druckserver von überall her konfigurieren
- ausreichende Sicherheit (Authentifizierung der Nutzer, Verschlüsselung der Druckdaten...) für Drucken über das Internet
- Zusammenspiel mit modernen Internet-Konzepten (Web-Browsing, Verzeichnisdienste, . . .)

Gerade an den letzten beiden Punkten ist zu sehen, daß die *Printer Working Group* bestrebt war, das Printing in eine EDV-Welt zu integrieren, in der viele Dienste zusammenarbeiten und voneinander abhängig sind.

### 1.3 Realisierung des IPP

Das Rad sollte nicht noch einmal neu erfunden werden. So wird für die Kommunikation das HTTP<sup>2</sup>-Protokoll genutzt. Zwischen Client und Server werden XML<sup>3</sup>-Pakete hin- und hergeschickt, also muß die Kommunikation bidirektional sein. In diesen Paketen können nicht nur Status-Informationen enthalten sein, sondern auch Text-, Bild- oder Treiberdaten.

Mittels HTTPS kann eine verschlüsselte Verbindung hergestellt werden, und da IPP mit dem HTTP so eng verwandt ist, können Webbrowser direkt auf den Dienst zugreifen. Sie müssen nur den **Port 631** ansprechen, der für das IPP reserviert ist. Alternativ ist in vielen Browsern auch eine eigene Methode realisiert: *ipp://*

---

<sup>1</sup>Printing Description Language

<sup>2</sup>Hypertext Transfere Protocol

<sup>3</sup>Extensible Markup Language

## Kapitel 2

# Das Common Unix Printing System

CUPS selbst ist die am weitesten fortgeschrittene Implementierung des Internet Printing Protokoll. Entwickelt wurde und wird das CUPS von der Firma *Easy Software Products*. Die Software selbst steht aber unter der *GNU General Public License*(GPL) und kann auf Rechnern mit Linux oder verschiedenen UNIX-Betriebssystemen wie AIX, Solaris und Compaq/HP übersetzt und installiert werden.

Neben einer großen Anzahl an Treibern für viele verschiedene Drucker – in Form von PPD<sup>1</sup>-Dateien – bringt CUPS einen eigenen LPR-Daemon mit, um die Aufgaben des älteren Printing Protokolls LPR/LPD übernehmen zu können.

### 2.1 Konzept

Ist CUPS als Druckdienst auf einem Rechner installiert, läuft im Hintergrund ein Daemon, der auf dem Port 631 horcht und eingehende Service-Anfragen entgegennimmt.

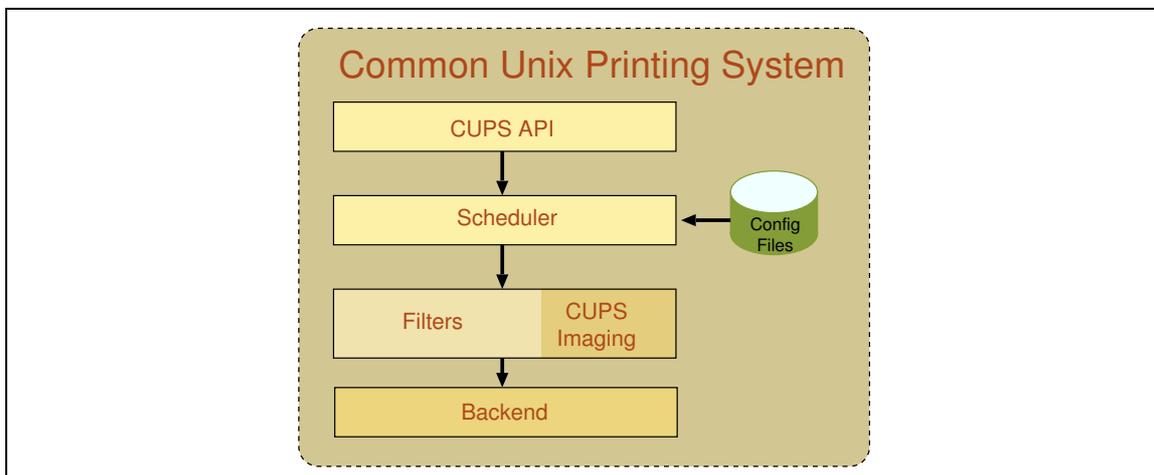


Abbildung 2.1: CUPS – Ablaufdiagramm

Wie in Abbildung 2.1 zu sehen ist, besitzt CUPS eine sogenannte API<sup>2</sup>, in der verschiedene Schnittstellen existieren, mit deren Hilfe Jobs an das *Common Unix Printing System* übergeben werden können. Im weiteren kümmert sich ein Scheduler darum, daß sie bearbeitet und aufbereitet werden. Dabei werden die verschiedensten Dateiformate (Text,Bilder,PDF,HTML, ...) erkannt, und mit entsprechenden Filtern in PostScript umgewandelt. Zusätzlich werden die eingestellten Druckeroptionen, wie z.B. Duplexdruck, Auflösung oder Papiergröße, eingefügt und eventuell konfigurierte

<sup>1</sup>Postscript Printer Description

<sup>2</sup>Application Programming Interface

Trennblätter mitgeschickt. Mittels sogenannter Backends werden die Jobs letztendlich zum eigentlichen Drucker geschickt.

Einfluß auf diesen Ablauf kann via verschiedener Konfigurationsdateien genommen werden. Siehe Abschnitt 2.4.6 auf Seite 15.

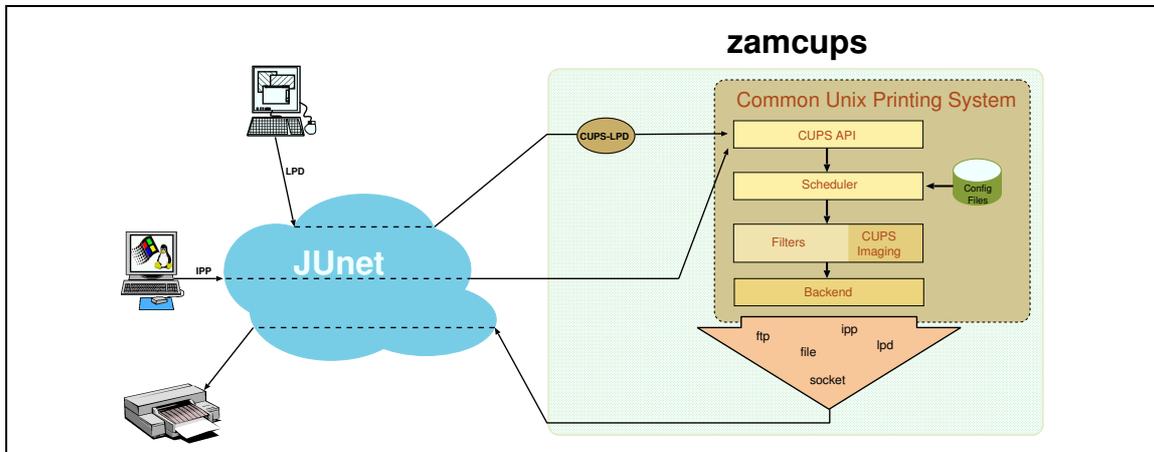


Abbildung 2.2: Jobverlauf

In Abbildung 2.2 wird der Ablauf eines Druckauftrags gezeigt. Windows/Linux schicken über das Netzwerk einen Job zum Printserver *zamcups* – IPP-Port 631. Der Daemon nimmt ihn entgegen, bereitet ihn auf und schickt ihn wieder über das Netzwerk zum Drucker.

Ältere Systeme, die das IPP noch nicht kennen, können das LPR/LPD Protokoll nutzen. Dabei werden die Daten über das Netz zum Port 515 des Printservers geschickt. Dort nimmt der LPD-Mini-Daemon *cups-lpd* den Job entgegen und gibt ihn weiter an das CUPS. So wird eine gewisse Abwärtskompatibilität gewährleistet.

## 2.2 Installation von CUPS 1.1.23 auf einem SuSE Linux Server

1. **Anmerkung:** Bei der SuSE-Version handelt es sich um SLES9 (SuSE Linux Enterprise Server)
2. **Anmerkung:** Für die Installation und Administration des CUPS-Dienstes werden Administratorrechte (root) benötigt.

In vielen Linux-Distributionen ist inzwischen CUPS als Standard Printing-System voreingestellt. Es gibt zwei Arten der Grundkonfiguration:

- komplett
- Client-only

Bei der *Client-only* Variante läuft kein eigener Dienst auf dem lokalen System mehr, sondern alle Aufträge werden direkt an einen Server-Rechner weitergegeben. Diese Art der Installation wird von den Herstellern aus Sicherheitsgründen empfohlen, denn ein Dienst, der nicht läuft, kann auch nicht angegriffen werden.

Sobald jedoch ein Rechner direkt auf einen Drucker zugreifen möchte, oder es einen Printserver im Intranet gibt, der die IPP-Anforderungen nicht entsprechend realisiert hat (siehe Abschnitt 1.2 auf Seite 3), muß ein Komplettsystem von CUPS installiert werden.

Nach der Installation sollte der Dienst laufen. Der Befehl `rccups status ↔` kontrolliert das. Falls er nicht läuft, wird er mit `rccups start ↔` gestartet. Alle *Start-up* Skripte liegen im Verzeichnis `„/etc/init.d/“`. Mit `insserv cups ↔` kann der CUPS-Dienst per Kommando in den Bootvorgang integriert werden. Alternativ kann bei SuSE die Konfiguration auch per *Yast2* → *Dienste* durchgeführt werden.



Abbildung 2.3: Web-Interface des CUPS-Dienstes

Sobald der Dienst läuft, kann in einem Webbrowser über die Adresse `http://localhost:631` auf den CUPS-Dienst des lokalen Rechner zugegriffen werden (siehe Abbildung 2.3). Für „localhost“ kann auch jeder andere Rechner-Name eines IPP-Printservers angegeben werden. So wird die Anforderung der IPP-Spezifikation realisiert (s. Abschnitt 1.2 auf Seite 3), den Server *remote* zu konfigurieren sowie die Jobs vom Client aus zu überwachen. Viele Browser unterstützen auch mit der *ipp*-Methode den direkten Zugriff auf Port 631.

Diese Webseite ist Ausgangspunkt um

1. neue Drucker(-klassen) zu installieren, konfigurieren und löschen (Administration),
2. Drucker(-klassen) zu starten/stoppen, Jobs anzuhalten oder zu löschen (Operating) und
3. den Status des Druckers/Jobs abzufragen (Benutzer).

Wer mit welchen Privilegien auf welche Ressourcen zugreifen darf, wird in der Konfigurationsdatei „`./etc/cups/cupsd.conf`“ geregelt (siehe Abschnitt 2.4.1 auf Seite 11).

#### **Begriffserklärung: Druckerklasse**

*Eine Druckerklasse besteht aus mehreren gleichwertigen Druckern, die unter **einem** Namen zusammengefaßt sind. Jobs, die an eine Druckerklasse geschickt werden, werden vom CUPS-Scheduler auf den nächsten freien Drucker der Klasse geschickt. (Load Balancing, High Availability)*

## 2.3 Installation und Konfiguration eines neuen Druckers

### 2.3.1 Web-Interface

Von der CUPS-Startseite (Abbildung 2.3) kommt man über den Link *Einrichtungsaufgaben* in das Administrationsmenü. Dort gibt es einen Button *Drucker hinzufügen*, um einen Drucker in 5 Schritten zu installieren (siehe Abbildung 2.4 auf der nächsten Seite).

Im ersten Schritt wird zwingend der Name (ohne Blanks) verlangt, optional können noch ein Standort und Eigenschaften angegeben werden. Als zweites wird ausgewählt, wie der Drucker angesprochen wird (entnimmt man dem Drucker-Handbuch). In dieser Auswahl werden alle sogenannten Backends aufgelistet, die CUPS kennt. Wie man weitere Backends erstellt und hinzufügt, wird in Abschnitt 2.5 auf Seite 15 erläutert.

Im dritten Schritt müssen die Kommunikations-Daten des Druckers angegeben werden: In der Regel den Hostnamen und in diesem Beispiel den Queuenamen „PS“ des Druckers (siehe Drucker-

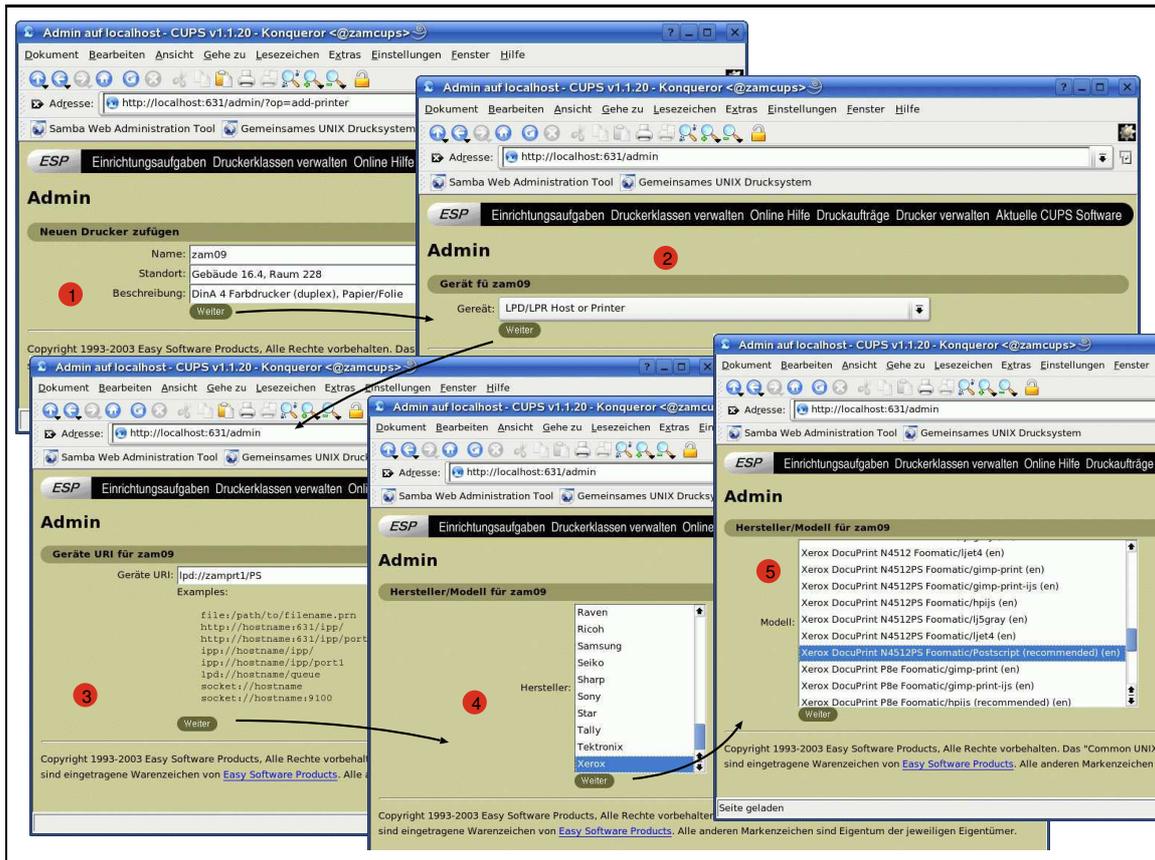


Abbildung 2.4: Druckerinstallation via Web-Interface

Handbuch). Als letztes muß der passende Treiber ausgewählt werden. CUPS selbst bringt schon eine große Anzahl an Treibern mit.

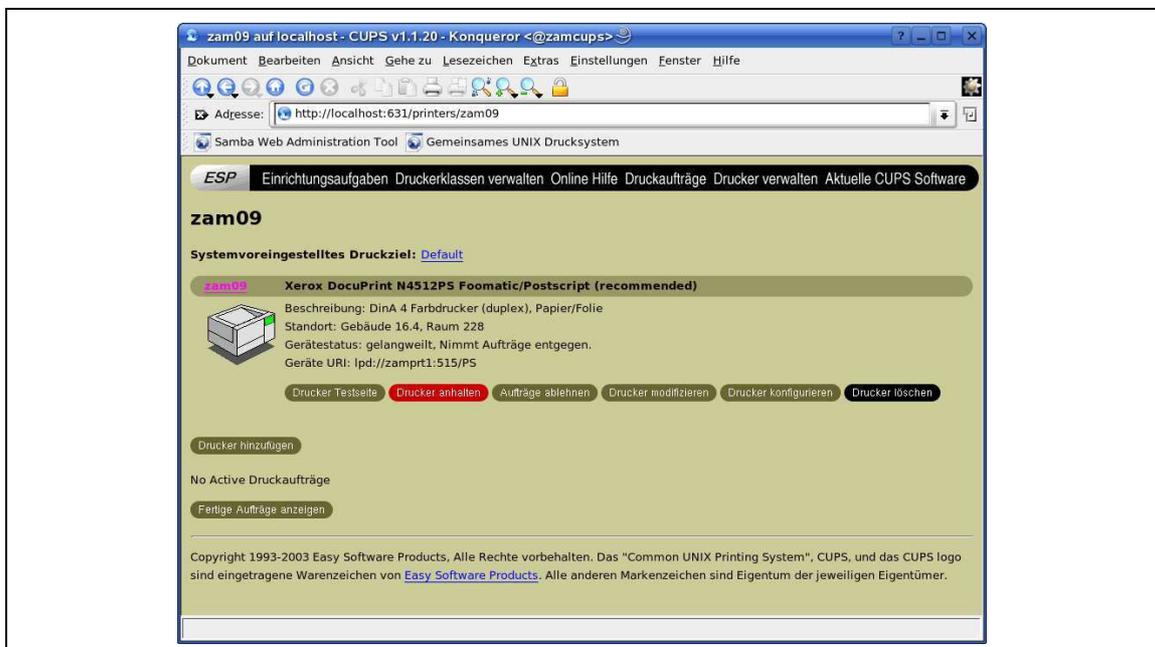


Abbildung 2.5: Status des neu installierten Druckers

Ist die Installation abgeschlossen, ist der Drucker einsatzbereit. Über das Web-Interface können sich die Benutzer jetzt den Status des Druckers anzeigen lassen (siehe Abbildung 2.5).

Dort wird die Schaltfläche *Drucker konfigurieren* angeboten. Darüber gelangt man zu einer Oberfläche, in der alle optionalen Parameter eines Druckers vom Administrator voreingestellt werden

können. Typische Eigenschaften sind Papiergröße, Auflösung, Papierfach, Papierart und Papierfach. Dazu kommen optionale Ausstattungsmerkmale des Druckers wie z.B. eingebaute Festplatte, Speichergröße und Duplexeinheit. Alle diese Informationen sind treiberabhängig und damit von Drucker zu Drucker verschieden.

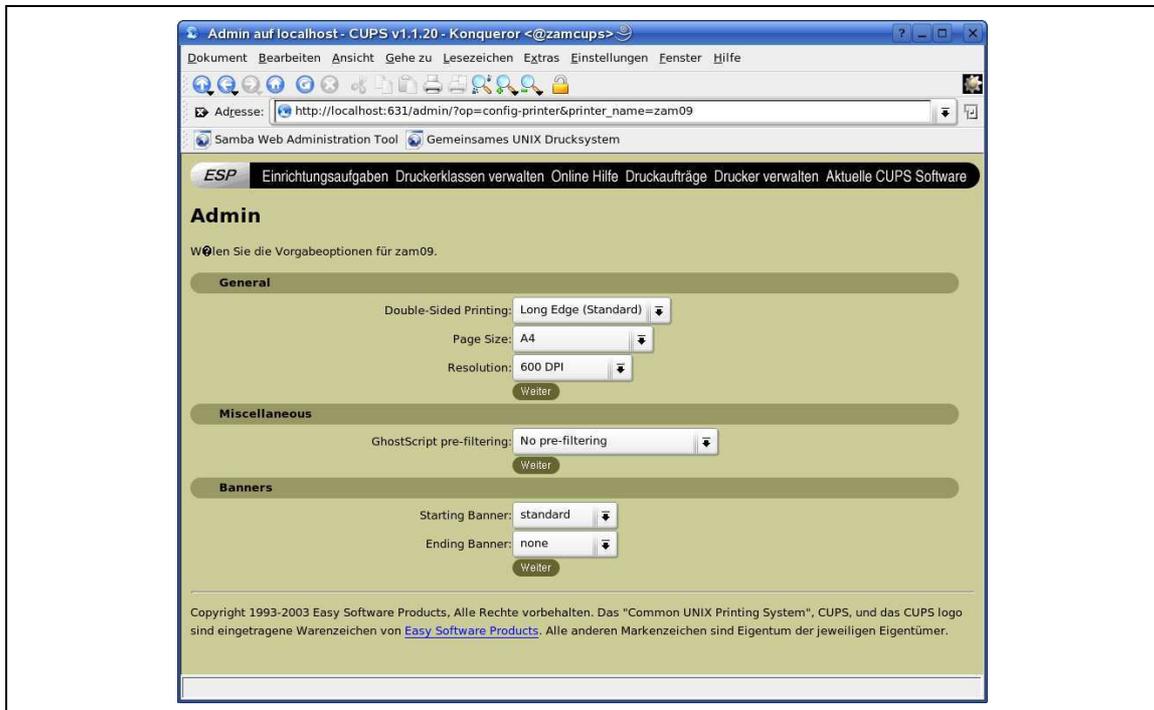


Abbildung 2.6: Konfiguration des Druckers

Einzig die Auswahl einer Banner-Page vor bzw. nach dem Druckjob sind modellunabhängig. In Abschnitt 2.6 auf Seite 16 wird auf das Thema *Bannerpage* genauer eingegangen.

Neben diesem Menü gibt es für *Foomatic/Postscript* Drucker noch die Möglichkeit, ein Prefiltering einzuschalten.

### **Begriffserklärung: Foomatic**

*Dabei handelt es sich um ein Paket aus Software-Tools und einer Datenbank aus freien und getesteten Treibern für Drucker. Das Foomatic wird in verschiedenen Printing System genutzt, u.a. in CUPS.*

### 2.3.2 Installation mit System-Kommandos

Neben dem benutzerfreundlichen Web-Interface gibt es – wie in der UNIX-Welt üblich – auch Kommandozeilen-basierte Schnittstellen, die oft mehr Möglichkeiten bieten, schneller sind und leicht in Skripts integriert werden können.

Für die Installation und Konfiguration eines Druckers gibt es den „lpadmin“- und „lpoptions“-Befehl.

#### **lpadmin:**

Der *lpadmin*-Befehl installiert einen neuen Drucker komplett mit einem Aufruf, es können auch einzelne Eigenschaften wie „Ort“ oder „Modell/Treiber“ geändert werden. Alle möglichen Optionen sind in Tabelle 2.1 auf der nächsten Seite aufgelistet.

Eine Alternative zu der Installation, wie in Abbildung 2.4 auf der vorherigen Seite beschrieben, wäre folgender Befehl:

-p <Name>	→Name der Druckerqueue
-E	→Der Drucker wird direkt gestartet (enable)
-L <location>	→Beschreibung des Standortes des Druckers
-D <info>	→ Informationen über den Drucker (A4/A3, Farbe/sw, duplex/simplex,...)
-m <model>	→Druckertreiber aus der CUPS-Datenbank
-P <PPD-Datei>	→eigener Druckertreiber (Original vom Hersteller)
-v <Device>	→ Schnittstelle, über die der Drucker angesprochen wird (siehe dazu auch das CUPS-Administrator Handbuch)
-o <options>	→Druckertreiber-Optionen

**Tabelle 2.1:** Optionen des *lpadmin*-Befehls

```
lpadmin -p zam09 -E -L "Gebäude 16.4, Raum 228" \
-D "DinA4 Farbdrucker (duplex), Papier/Folie" \
-m Xerox/DocuPrint_N4512PS-Postscript.ppd.gz3 \
-v lpd://zamprt1/PS ↔
```

Der Befehl ermöglicht es, sehr einfach und schnell einzelne Parameter eines Druckers zu ändern, ohne sich durch das komplette Menü hangeln zu müssen. So würde das Kommando

```
lpadmin -p zam09 -L "Gebäude 16.3, Flur West" ↔
```

nur die Ortsbeschreibung ändern.

Für die Konfiguration der Drucker (via Web siehe Abbildung 2.6 auf der vorherigen Seite) steht dem Befehl „*lpadmin*“ die Option „-o“ zur Verfügung. Der Befehl

```
lpadmin -p zam09 -o Duplex=DuplexNoTumble ↔
```

aktiviert standardmäßig den Duplexdruck. Welche Optionen der Drucker alle hat, gibt der „*lpoptions*“ Befehl an.

### lpoptions:

Mit dem „*lpoptions*“-Befehl kann der Administrator systemweit, aber jeder Benutzer für sich selber, die Konfiguration der Drucker an seine Bedürfnisse anpassen.

-p <Name>	→Drucker, der manipuliert wird
-E	→Aktiviert Verschlüsselung (enable)
-d <Name>	→setzt den Default Drucker
-l	→listet alle Optionen eines Druckers auf
-x <Name/Instanz>	→löscht alle Optionen eines Druckers / löscht eine Instanz
-o <options>	→setzt Druckertreiber-Optionen

**Tabelle 2.2:** Optionen des *lpoptions*-Befehls

Um die Optionen eines Druckers zu setzen, müssen sie erst bekannt gemacht werden. Es gibt zwar eine Reihe von Optionen, die den meisten Druckern gemeinsam sind, zusätzlich gibt es oft Spezialoptionen, die die Hersteller selbst definiert haben. Mit dem Kommando

```
lpoptions -p zam09 -l ↔
```

werden alle Optionen des Druckers „*zam09*“ aufgelistet (siehe Abbildung 2.7 auf der nächsten Seite).

Diese Informationen kann der Benutzer nutzen, um den Drucker für seine Anforderungen einzustellen. Z.B. wird er Befehl

```
lpoptions -p zam09 -o Resolution=600x600dpi ↔
```

die Auflösung auf 600 DPI setzen. Die gerade aktive Option ist mit einem „\*“ markiert.

<sup>3</sup>mit dem Befehl `lpinfo -m ↔` werden alle verfügbaren Treiber aufgelistet

```
Duplex/Double-Sided Printing: *DuplexNoTumble DuplexTumble None
Notcapable
PageSize/Page Size: *A4 Letter 11x17 A3 A5 B5 Env10 EnvC5 EnvDL
EnvISOB5 EnvMonarch Executive Legal
PageRegion/PageRegion: A4 Letter 11x17 A3 A5 B5 Env10 EnvC5 EnvDL
EnvISOB5 EnvMonarch Executive Legal
Resolution/Resolution: *default 150x150dpi 300x300dpi 600x600dpi
PreFilter/GhostScript pre-filtering: EmbedFonts Level1 Level2 *No
```

Abbildung 2.7: lpoptions -p zam09 -l

Sollen die Optionen generell zentral auf dem Printserver gesetzt werden, muss dafür der „lpadmin“-Befehl genutzt werden (siehe Abschnitt 2.3.2 auf Seite 9).

Ein weiteres schönes Feature bei CUPS ist das Erstellen von *Druckerinstanzen*. Angenommen, ein Drucker kann sowohl Papier – DinA4 und DinA3 – als auch Folien bedrucken. Dann müßte der Benutzer jedesmal die entsprechende Option beim Drucken mit angeben, denn es kann nur eine Eigenschaft als Default voreingestellt sein. Deswegen bietet CUPS die Möglichkeit, für einen Drucker eine weitere Instanz zu definieren, für die ganz andere Default-Optionen gesetzt sind als beim Original.

So wird z.B. mit dem Befehl

```
lpoptions -p zam09/pd -o Duplex=True ↵
```

eine Druckerinstanz von „zam09“ mit dem Namen „zam09/pd“ angelegt, bei der jeder Job, der mit

```
lpr -P zam09/pd <Job> ↵
```

abgeschickt wird, automatisch doppelseitig gedruckt wird.

#### **Begriffserklärung: Druckerinstanz**

*Eine Druckerinstanz ist ein Alias-Name eines Druckers, bei der spezielle Optionen eingestellt sind. Der Benutzer kann den Alias-Namen wie einen normalen Drucker nutzen.*

## 2.4 CUPS-Konfigurationsdateien

### 2.4.1 /etc/cups/cupsd.conf

Die „cupsd.conf“-Datei ist das Herzstück der Konfiguration von CUPS; mit ihr wird der Daemon eingestellt. Die Beschreibung aller Optionen sind im Administrator-Handbuch von CUPS nachzulesen(→[L1]). Hier werden nur die Optionen behandelt, die beim Print-Server des ZAM von der Default-Konfiguration abweichen.

**Der Befehl `rccups restart` ↵ startet den Daemon mit der neuen Konfiguration!**

Die Datei „cupsd.conf“ kann in zwei Abschnitte unterteilt werden.

#### 2.4.1.1 Allgemeine Parameter

#### 2.4.1.2 Sicherheitseinstellungen

Zugriff auf den CUPS-Dienst geschieht per Web-Adressen über den Port 631. In der Datei „/etc/cups/cupsd.conf“ können für jede Adresse – also jede Ressource – die Zugriffsrechte explizit gesetzt werden.

Für jede Adresse muß ein eigener Abschnitt nach folgendem Muster existieren:

RemoteRoot remote-root	→ Setzt UserID-String für „root“ von Remote-Systemen auf „remote-root“
MaxClients 1000	→ Anzahl Clients, die gleichzeitig auf den Server zugreifen dürfen.
Browsing Off	→ Server nimmt nicht am Browsing via Broadcast teil.
BrowseAllow 134.94.*	→ Alle Rechner aus dem Forschungszentrum dürfen den CUPS-Daemon ansprechen
BrowseDeny All	→ Alles andere wird abgelehnt.
BrowseOrder deny,allow	→ Alles wird verboten, was nicht explizit erlaubt ist.
SystemGroup lp	→ Printer-Administrator müssen zur Gruppe „lp“ gehören
HostNameLookups On	→ Namensauflösung eingeschaltet
Port 631	→ IPP-Port für den Dienst

**Tabelle 2.3:** Auswahl an Parametern der „cupsd.conf“-Datei

```
<Location />
Order Deny,Allow
Deny From All
Allow From 127.0.0.1
Allow From 127.0.0.2
Allow From @LOCAL
Allow From 134.94.*
</Location>
```

Bei dieser Konfiguration dürfen alle Rechner aus dem Forschungszentrum auf die Startseite des CUPS-Servers zugreifen („http://zamcups:631“).

Der Drucker „zam09“ soll nur den ZAM-Mitarbeitern zur Verfügung stehen. Eine geeignete Konfiguration wäre:

```
<Location /printers/zam09>
Order Deny,Allow
Deny From All
Allow From 134.94.168.255/255.255.248.0
</Location>
```

Außerdem dürfen nur privilegierte Benutzer die Drucker administrieren:

```
<Location /admin>
AuthType Digest
AuthClass Group
AuthGroupName lp
Order Deny,Allow
Deny From All
Allow From @LOCAL
Allow From 134.94.168.42
</Location>
```

### 2.4.1.3 Einschub: Printer-Administrator Account

Der Zugriff auf bestimmte Ressourcen kann über eine Benutzer-Autorisierung erfolgen. Es werden zwei Arten angeboten:

1. *Basic Authentication*
2. *Digest Authentication*

Unabhängig von der Methode geht das Paßwort im Klartext über die Leitung, es sei denn, die IPP-Kommunikation wird insgesamt verschlüsselt. Deswegen ist von der *Basic Authentication* ab-

zuraten, da dort die Accounts und Passörter des Linux-Systems genutzt werden. Bei der *Digest Authentication* wird vom CUPS eine eigene Paßwort-Datei „/etc/cups/passwd.md5“. Selbst wenn jemand diese Paßwörter mitliest, hat er noch keinen Zugriff auf einen Benutzer-Account des Linux-System.

Das Kommando um den neuen Benutzer „prtoper“ hinzuzufügen lautet:

```
lppasswd -g lp -a prtoper ↵
```

*prtoper* gehört der Gruppe „lp“ an. Wenn die Sicherheitseinstellung also wie folgt definiert sind,

```
...
AuthClass Group
AuthGroupName lp
...
```

dann können alle Mitglieder der Gruppe *lp* auf die angegebene Ressource zugreifen.

```
lppasswd -x prtoper ↵
```

löscht den Benutzer wieder.

### 2.4.2 /etc/cups/client.conf

In der Datei „client.conf“ wird der Rechner eingetragen, der für das System das Spooling übernimmt. In der Regel ist es der lokale Rechner (localhost).

Bei einer einheitlichen Druckinfrastruktur über CUPS ist eine solche Konfiguration auf den Clients sinnvoll. CUPS wird in diesem Fall nur in der „Client-Only“ Version installiert. Jede Anfrage an das Printing wird weitergeleitet an den Server.

Sind noch andere Printserver-Systeme im Einsatz – z.B. LPR/LPD-Server – müssen dessen Drucker lokal auf den Clients installiert werden und das Spooling übernehmen die Rechner selber. Ein Drucker kann auch direkt an einem Rechner angeschlossen sein. Dann dient der Client selber als Printserver für dieses Gerät. In beiden Fällen hat das zur Folge, daß auf dem Client ein kompletter CUPS-Dienst installiert werden muß.

Für die „Client-Only“ Konfiguration muss folgendes in die „client.conf“-Datei eingetragen werden:

```
ServerName zamcups.zam.kfa-juelich.de
```

### 2.4.3 /etc/cups/printers.conf

Für jeden installierten Drucker wird in der Datei „printers.conf“ ein eigener Eintrag erstellt.

```
<Printer zam09>
Info DinA4 Farbdrucker (duplex), Papier/Folie
Location Gebäude 16.4, Raum 228
DeviceURI lpd://zamprt1/PS
State Idle
Accepting No
JobSheets none none
QuotaPeriod 0
PageLimit 0
KLimit 0
</Printer>
```

**Abbildung 2.8:** Einträge zum Drucker „zam09“ in „printers.conf“

Dort werden alle Informationen zu einem Drucker festgehalten, die nicht den Treiber direkt betreffen. Also alles, was beim „lpadmin“-Befehl per Optionen angegeben wird (s. Abschnitt 2.1 auf Seite 10).

### 2.4.4 /etc/cups/lpoptions

Der Administrator des Printservers gibt bei jedem Drucker eine Standard-Konfiguration vor. Möchte ein Endbenutzer an einem Client andere Optionen des Druckers nutzen, kann er die entsprechende Option bei jedem Job mit angeben.

Er hat auch die Möglichkeit, die Default-Einstellungen mit dem „lpoptions“-Befehl zu ändern, oder sich Druckerinstanzen einzurichten, die unter dem neuen Namen die gewünschten Optionen direkt voreingestellt haben. (siehe Abschnitt 2.2 auf Seite 10)

Diese Informationen werden für jeden Benutzer in seinem Heimatverzeichnis in der Datei „.lpoptions“ oder vom Administrator systemweit in der Datei „/etc/cups/lpoptions“ gespeichert.

```
Default zam08/pd
Dest zam02/cd Duplex=DuplexNoTumble
Dest zam02/d3 Duplex=DuplexNoTumble PageSize=A3
```

**Abbildung 2.9:** Beispiel-Einträge in der „lpoptions“-Datei

In Abbildung 2.9 ist ein Beispiel für eine „lpoptions“-Datei. Dort werden die beiden Druckerinstanzen „zam02/cd“ und „zam02/d3“ definiert. Die Druckerinstanz „zam08/pd“ ist der Default-Drucker.

### 2.4.5 /etc/cups/classes.conf

CUPS bietet die Möglichkeit, mehrere Drucker unter einem Namen zusammenzufassen. Man bezeichnet das als *Druckerklasse*. Es verbessert die gleichmäßige Auslastung und erhöht die Ausfallsicherheit: Der Job wird auf den ersten verfügbaren (freien) Drucker geschickt.

Die Informationen zu einer Druckerklasse – z.B. welche Drucker dazugehören - werden in der Datei „classes.conf“ gespeichert.

```
<Class zam>
Info Druckerpool DinA4 s/w
Location ZAM - I/O-Raum
State Idle
Accepting Yes
JobSheets none none
Printer zam08
Printer zam18
QuotaPeriod 0
PageLimit 0
KLimit 0
</Class>
```

**Abbildung 2.10:** Klasse „zam“ in „classes.conf“

Eingerichtet werden Klassen mit dem *lpadmin*-Befehl oder per Web-Interface (siehe Abschnitt 2.3 auf Seite 7).

Neben den vom Administrator definierten Druckerklassen in der „classes.conf“-Datei gibt es noch „implizite“ Klassen. Weitere Informationen dazu sind der CUPS-Dokumentation zu entnehmen.[L1]

### 2.4.6 /etc/cups/mime.types und /etc/cups/mime.convs

CUPS bietet eine automatische Format-Erkennung z.B. für *PDF*-, *JPEG*- oder *Text*-Dateien. Fast alle Dateien fangen für ihren Typ charakteristisch an. Welche CUPS kennt, ist in der Datei „`/etc/cups/mime.types`“ definiert.

```
application/pdf pdf string(0,%PDF)
```

**Abbildung 2.11:** PDF Dateien beginnen mit PDF (Zeile aus „`mimes.types`“)

In Abbildung 2.11 steht beschrieben, daß es einen Dateityp „PDF“ gibt. Die ersten Zeichen in so einer Datei müssen „%PDF“ lauten.

Der CUPS-Scheduler entscheidet anhand der Datei „`/etc/cups/mime.convs`“, wie Dateien eines erkannten Typs behandelt werden müssen, welches Programm daraus ein PostScript erstellt. Z.B. werden PDF-Dateien mit dem Programm *pdftops* in PostScript umgewandelt (Siehe Abbil-

```
application/pdf application/postscript 33 pdftops
```

**Abbildung 2.12:** pdftops wandelt PDF in PostScript um (Zeile aus „`mimes.convs`“)

dung 2.12).

## 2.5 CUPS-Backend – Kommunikation mit den Druckern

In diesem Abschnitt wird genauer darauf eingegangen, wie der CUPS-Daemon mit den eigentlichen Druckern kommuniziert. Bei der Installation der Drucker (s. Abschnitt 2.3 auf Seite 7) wurden die sogenannten *Backends* erwähnt. Der Dokumentation des Druckers bzw. dem Administrator-Handbuch von CUPS ist zu entnehmen, welches Backend für welchen Drucker eingesetzt werden soll.

Die Backend-Programme aus Tabelle 2.4 liegen im Verzeichnis „`/usr/lib644/cups/backend`“.

canon	→Backend für Canon Drucker
epson	→Backend für Epson Drucker
ipp/http	→Backend für Drucker, die IPP nutzen (z.B. andere IPP-Printserver)
lpd	→LPR/LPD-Backend für entsprechende Drucker oder Printserver
parallel	→Backend für Drucker am Parallel-Port
pipe	→ Output-Umlenkung auf Kommando oder in Datei, entspricht dem Pipe-Symbol in der UNIX/Linux-Shell
scsi	→Backend für SCSI-Drucker
serial	→Backend für serielle-Drucker
smb	→Schnittstelle aus Samba für Windows-Drucker (Windows-Share)
socket	→Backend für drucker mit Socket-Kommunikation, z.B. HP-Drucker
usb	→Backend für Drucker am USB-Port
novell	→ Inoffizielles Backend von Novell/SuSE, Zugriff mit User-ID + Passwort auf eine entfernte Druckerqueue

**Tabelle 2.4:** Backends von CUPS

<sup>4</sup>nur auf einem 64-Bit System, sonst ohne die „64“

### 2.5.1 Erstellen eigener Backends

Es ist möglich, eigene Backends zu erstellen. Dazu muss man wissen, wie die Backends vom CUPS-Daemon aufgerufen werden, und wie der Return-Wert bzw. die Ausgabe auszuschaun hat.

#### 1. Aufruf des „Backend“ (Syntax)

Einem Backend werden 5 bzw. 6 Argumente vom CUPS übergeben:

```
<backend> <Job-Id> <User> <Titel> <Kopien> <Optionen> <Datei> ←
```

Das „Optionen“-Feld wird zur Zeit noch nicht genutzt, und falls keine Datei angegeben ist, wird der Job von Standard-Eingabe erwartet. Aufgerufen werden die Backends vom CUPS-Scheduler.

**Wichtig:** Wird das Backend ohne Parameter aufgerufen, muß es Informationen in einem speziellen Format über sich preisgeben. Diese werden vom CUPS erwartet, damit das Backend erkannt und eingesetzt werden kann.

Beispiel bei dem „lpd“-Backend:

```
network lpd "Unknown" "LPD/LPR Host or Printer"
```

Wird das Backend mit allen Optionen aufgerufen, um einen Job zum Drucker zu schicken, erwartet CUPS einen Rückgabewerte

= 0 für Erfolg,  
 ≠ 0 für einen Fehler.

Beispiel für ein selbstgeschriebenes Backend ist das „ftp“-Backend im Anhang.

Auch das Tool *tea4cups* (siehe Abschnitt 4.2 auf Seite 29) bringt ein eigenes Backend mit.

## 2.6 Bannerpages in CUPS

CUPS bietet die Möglichkeit, Deckblätter zu definieren. Ein sogenanntes *Starting Banner*, welches vor dem Job zum Drucker geschickt wird, und ein *Ending Banner*, das nach einem Job gesendet wird. Mit „lpadmin“ bzw. per Web-Interface (siehe Abschnitt 2.3 auf Seite 7) können die Seiten vom Administrator voreingestellt werden.

Alle Seiten, die CUPS kennt, liegen im Verzeichnis „/usr/share/cups/banners“.

Bei jedem Job, den ein Benutzer schickt, kann er die voreingestellten Banner-Seiten nach eigenem Ermessen ändern. Bedingung ist, es muß eine Seite sein, die der CUPS-Server kennt.

Leider bietet CUPS nicht die Möglichkeit – zumindest nicht in der aktuellen Version, das Deckblatt aus einem bestimmten Fach zu ziehen oder auf einem anderen (Farb-)Papier zu drucken. Um das zu lösen, muß das Deckblatt die Informationen selbst enthalten. So muß der Administrator für jeden Drucker ein eigenes Deckblatt erstellen.

### 2.6.1 Bannerpages selbst erstellen

Als Vorlage können die Deckblätter aus dem Verzeichnis „/usr/share/cups/banners“ genommen werden. Es können aber auch eigene erstellt werden. Wichtig ist, daß sie in der Seitenbeschreibungssprache PostScript geschrieben sind.

Am Anfang einer solchen Datei müssen spezielle PostScript-Befehle stehen (*setpagedevice*), die dem Drucker gewisse Anweisungen geben, um das gewünschte Papier auszuwählen. Typischerweise sind das:

- Papierfach (Manuelles Fach, Fach 1, ...)
- Papiergröße (DinA4: 842 pt x 595 pt)
- Papierart (Farbpapier, Karton, ...)

```

:
[ { %%BeginFeature:
<<
/DeferredMediaSelection true
/MediaType (Color)
/MediaClass (Color)
/ManualFeed false
/Colate false
/ImagingBBox null
/PageSize [595 842]
/TraySwitch false
/Duplex false
>> setpagedevice
%%EndFeature
} stopped cleartomark
:

```

**Abbildung 2.13:** Fachauswahl für eine *HP 8500* in einer Bannerpage

Leider handhabt jeder Druckerhersteller die Fachauswahl bei seinen Geräten anders. Deswegen ist es bei neuen Drucken manchmal schwierig, an diese Informationen heranzukommen. Ein Weg, wie man an die richtigen Befehle kommt, steht im Abschnitt 4.3 auf Seite 31.

Der zweite Teil der Bannerpage besteht aus den Elementen, die auf Papier erscheinen sollen. Für verschiedene Informationen sind Platzhalter vorgesehen, die vom CUPS automatisch mit Inhalt gefüllt werden. Alle möglichen Job-Informationen sind:

```

Benutzername:  {?job-originating-user-name}
Job-/Dateiname:  {?job-name}
DruckerQueue:  {?printer-name}
Zeitstempel:    {?time-at-creation}
Job-Nr.:        {?job-id}
Rechner-Name:  {?job-originating-host-name}
Druckkosten:   {?job-billing}
Job Priorität:  {?job-priority} von 100
Drucker URI:   {?job-printer-uri}

```

Nachteil dieser Methode ist, daß jeder Benutzer sich sein Deckblatt selber aussuchen kann. So besteht die Gefahr, daß eine Seite mit Druckerbefehlen zu einem Drucker geschickt wird, die dieser nicht verarbeiten kann.

Mit dem Tool *Tea4Cups* (siehe Abschnitt 4.2 auf Seite 29) besteht die Möglichkeit, Deckblätter vom Backend hinzufügen zu lassen, was der Benutzer nicht mehr beeinflussen kann.



## Kapitel 3

# Microsoft Windows und CUPS

Es gibt mehrere Wege, einen CUPS-Drucker unter Windows zu installieren. Aber egal, welchen man wählt, für jeden Drucker gilt, daß sich der Treiber nicht automatisch aktualisiert, wenn sich auf dem Server was ändert. Jedesmal muß der Treiber von Hand aktualisiert bzw. der Drucker ganz gelöscht und neu installiert werden.

Windows bietet von Haus aus zwei Möglichkeiten, auf Drucker eines CUPS-Printserver zuzugreifen,

- das LPR/LPD Protokoll und
- das IPP-Protokoll

In der Dokumentation von CUPS wird darauf hingewiesen, daß *Samba* die geeignete Software ist, Drucker von einem CUPS-Server in die Windows-Welt zu integrieren.

### 3.1 Drucken via LPR/LPD

Hierbei handelt es sich um die seit Windows NT klassische Art und Weise, UNIX/Linux-Drucker in das System einzubinden.

Zuerst müssen die *Weiteren Datei- und Druckdienste für das Netzwerk/Druckdienste für Unix* installiert werden. (Start → Systemsteuerung → Software → Windows-Komponenten), siehe Abbildung 3.1)

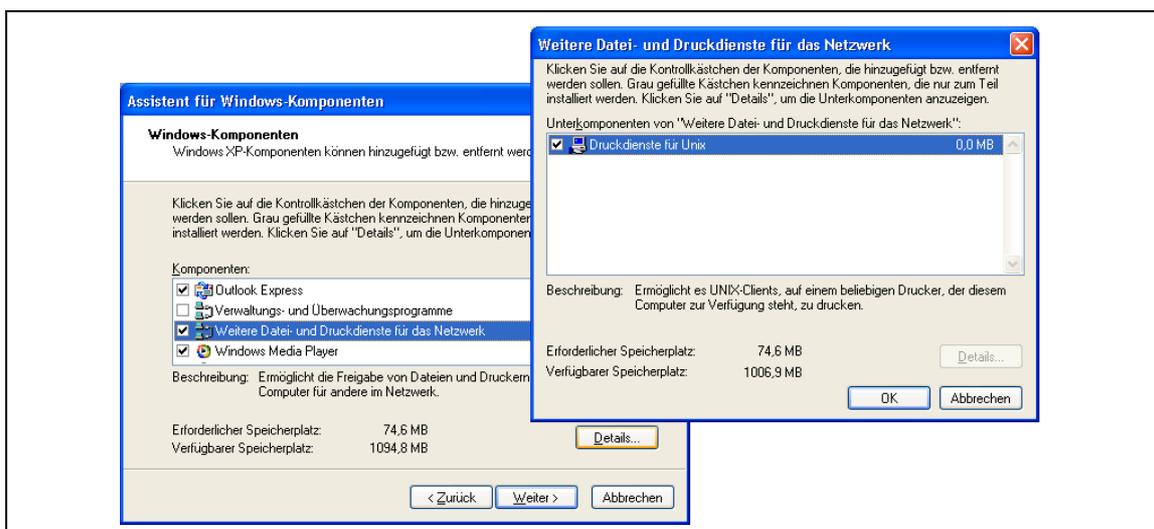


Abbildung 3.1: Unix Druckdienste für Windows

Damit steht bei der Einrichtung eines neuen Druckers ein neuer Port zur Verfügung: **LPR-Port**

Über „Neuen Drucker installieren“ (siehe Abbildung 3.2) wählt man aus, daß ein *lokaler Drucker* ohne *Plug & Play* eingerichtet werden soll. Für jeden Drucker dieser Art muß ein neuer *LPR-Port* angelegt werden. Es müssen der Server (oberes Feld), der den Drucker via LPR zur Verfügung stellt, und der Name des Druckers auf dem Server (unteres Feld) eingetragen werden. Die restlichen Schritte entsprechen denen einer Standard-Installation eines lokalen Druckers.

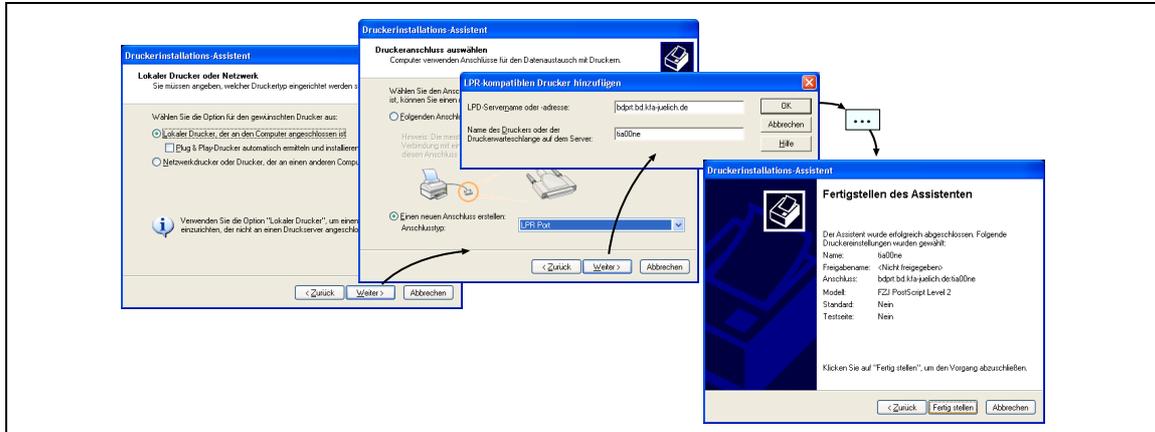


Abbildung 3.2: Windows LPR-Drucker einrichten

Auf diese Weise können Drucker eines CUPS-Server (und auch andere LPR/LPD fähiger Drucker im Netz) installiert werden. Zu beachten ist, daß ein Treiber benötigt wird, entweder einen generischen für alle Druckermodelle, oder für jedes Druckermodell seinen eigenen.

Im Forschungszentrum Jülich gibt es auf einem PC-Fileserver (*pcsrv.zam.kfa-juelich.de*) im Share *./public* ein Verzeichnis *./FZJ\_PS/WinXP*. Dort wird ein allgemeiner Druckertreiber bereitgestellt, der auch für diese Zwecke genutzt werden kann. [L4]

### 3.2 Drucken via IPP

Microsoft hat selbst an dem *Internet Printing Protocoll* mitgearbeitet, und hat in seinem Betriebssystem auch eine entsprechende Methode in den Versionen *Windows XP* bzw. *Windows 2000* implementiert.

IPP-Drucker fallen bei Windows unter die Netzwerk-Drucker, und werden nicht wie LPD/LPR-Drucker als „lokale Drucker“ behandelt.

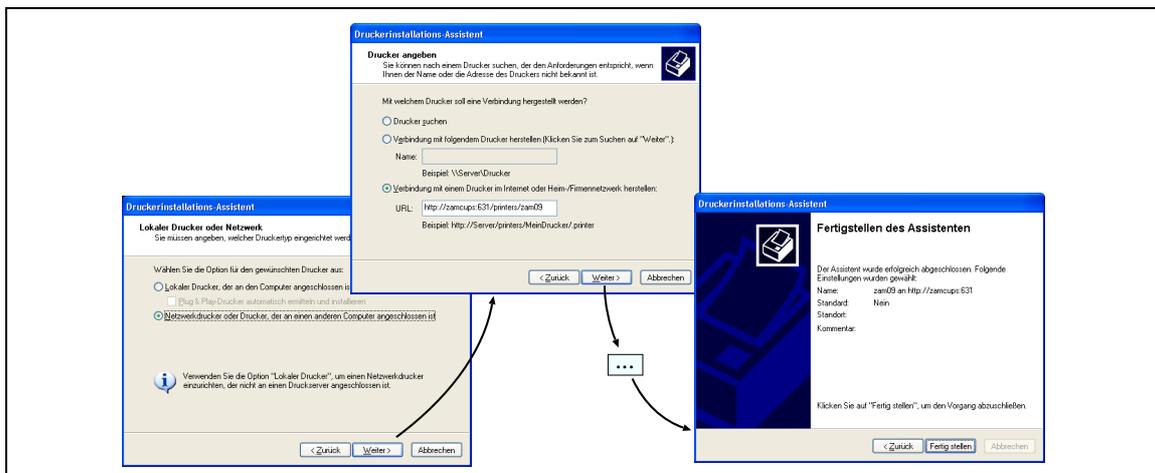


Abbildung 3.3: IPP-Drucker unter Windows installieren

In Abbildung 3.3 ist die Vorgehensweise bei der Installation eines IPP-Druckers aufgezeigt.

Im zweiten Fenster muß die Web-Adresse zum Drucker des CUPS-Servers angegeben werden, in

diesem Beispiel „`http://zamcups:631/printers/zam09`“. Zuletzt fragt das System nach dem geeigneten Treiber, und die Installation wird abgeschlossen.

### 3.3 CUPS und Samba

Samba ist eine Software, die Unix/Linux Ressourcen anderen Rechnern im Netz anbietet, so daß von einem PC unter Windows darauf zugegriffen werden kann. Der Autor, Andrew Tridgell, hat aus dem Acronym für das Übertragungsprotokoll SMB (Server Message Block) einen aussprechbaren Namen gemacht. Samba unterstützt eine Vielzahl von Clients und Servern.

In der Dokumentation zu Samba (→[L3]) gibt es ein eigenes Kapitel, das sich mit der Integration von Druckern eines CUPS-Servers beschäftigt. Abbildung 3.4 zeigt den Aufbau eines Printservers, der zusätzlich noch einen Samba-Server gestartet hat, über den die Windows-User die Drucker nutzen können.

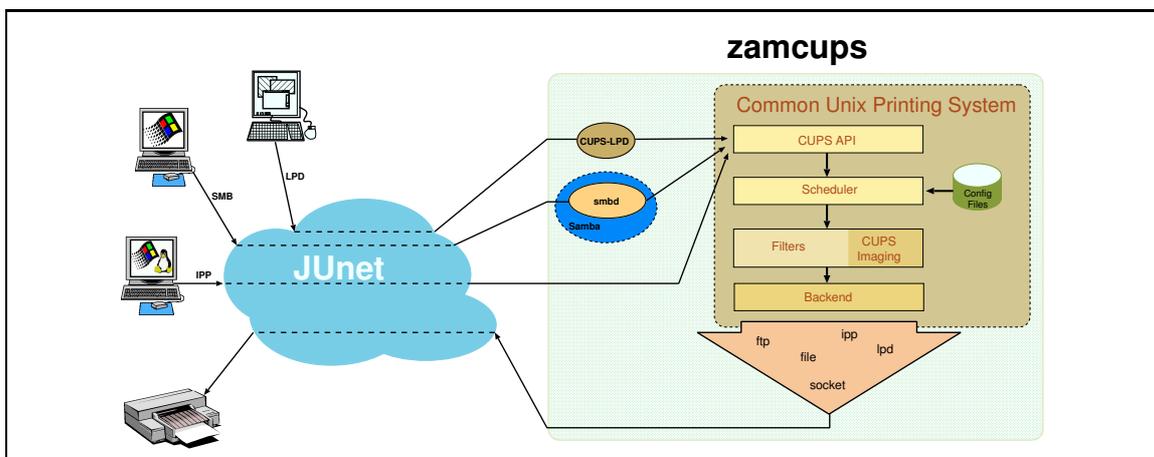


Abbildung 3.4: CUPS – Ablaufdiagramm mit Samba-Server

Der Vorteil von Samba ist, daß – bei richtiger Konfiguration – automatisch die Treiber mit exportiert werden, so daß der Windows Nutzer nichts weiter zu beachten hat.

Im folgenden wird eine Beispielkonfiguration beschrieben.

#### 3.3.1 Konfigurationsdatei `smb.conf`

Samba kann mehr, als nur die Drucker eines Unix-Systems für Windows zu exportieren. Da es sich aber um einen Printserver handelt, wird hier nur auf diesen Aspekt eingegangen.

Generell gilt, daß sich die Konfigurationsdatei „`smb.conf`“ in zwei Teile unterteilt:

- die „`global`“-Sektion, in der allgemeine Einstellungen vorgegeben werden, die zum einen als Default-Werte für die Shares gelten bzw. die nur den Samba-Dienst betreffen, wie z.B. der Rechnername, und
- die Abschnitte für die einzelnen Ressourcen.

##### 3.3.1.1 Die „`global`“-Sektion

In diesem Abschnitt (siehe Abbildung 3.5 auf der nächsten Seite) werden generelle Einstellungen für dem Samba-Dienst vorgenommen. Wichtigster Eintrag ist:

`map to guest = Bad Password`

Damit wird jeder Zugriff mit falschem Benutzer-Account/Passwort auf den Gast-Account umgelenkt, heißt, jeder darf als Gast die Shares nutzen.

```
[global]
    workgroup = ZAM
    server string = ZAM CUPS Server
    map to guest = Bad Password
    log level = 5
    log file = /var/log/samba/%m.log
    printcap name = cups
    os level = 0
    preferred master = No
    local master = No
    wins proxy = Yes
    wins server = eth0:134.94.80.84
    printer admin = root, prtoper
    hosts allow = 134.94.
    include = /etc/samba/dhcp.conf
```

Abbildung 3.5: „global“-Sektion der „smb.conf“ Datei

Mit dem Eintrag

```
hosts allow = 134.94.
```

wird der Zugriff auf ein Subnetz eingeschränkt. In diesem Fall dürfen nur Rechner aus dem Forschungszentrum Jülich die Shares nutzen.

Mit der Zeile

```
printer admin = root, prtoper
```

bekommen die Benutzer *root* und *prtoper* die nötigen Privilegien, um die Druckershare zu administrieren. Voraussetzung dafür ist, daß Samba die Benutzer-Accounts selber kennt. Dazu müssen sie in die Samba eigene Passwort-Datei eingetragen werden. Zum Beispiel wird mit dem Befehl

```
smbpasswd -a root ↵
```

der Benutzer „root“ hinzugefügt.

Weitere Optionen sind der Samba-Dokumentation zu entnehmen.

### 3.3.1.2 Die „printers“-Sektion

Mit diesem Abschnitt werden alle Drucker des Systems freigegeben. Als Share-Name erhalten die Drucker den Namen, den sie auch im Host-System haben.

```
[zam09]
    comment = Din A4 Farbe Papier
    path = /var/spool/samba
    guest ok = Yes
    hosts allow = 134.94.111.222
    printable = Yes
    printer name = zam09
```

Abbildung 3.6: Abschnitt für das Drucker-Share *zam09*

Besser ist es – wie in Abbildung 3.6 gezeigt – für jeden Drucker einen eigenen Abschnitt in der Konfigurationsdatei *smb.conf* zu definieren. So kann der Zugriff z.B. eingeschränkt werden („hosts allow = ...“) oder noch ein Kommentar mitgegeben werden. Man erhält mehr Flexibilität.

### 3.3.1.3 Das „print\$“-Share

In diesem Share liegen Treiber für alle exportierten Drucker, die bei einer Installation auf einem Windows-Client automatisch heruntergeladen und installiert werden.

```
[print$]
  comment = Printer Drivers
  path = /var/lib/samba/drivers
  write list = root, prtoper
  create mask = 0664
  directory mask = 0775
  guest ok = Yes
  browseable = No
```

Abbildung 3.7: Abschnitt für die Druckertreiber

Die Freigabe ist zwar für jeden lesbar, aber nicht sichtbar. Der Benutzer muß auch nicht wissen, wo welche Dateien stehen. Das erledigt Windows für ihn.

In Abbildung 3.7 sieht man, welches Verzeichnis als Freigabe dient („path = ...“). Damit Windows das Share als Treiber-Quelle akzeptiert, könnte die Verzeichnisstruktur wie folgt aussehen:

```
tree /var/lib/samba/drivers
|-- W32ALPHA
|-- W32MIPS
|-- W32PPC
|-- W32X86
|   |-- 2
|   |   |-- cups5.hlp
|   |   |-- cupsdrv5.dll
|   |   |-- cupsui5.dll
|   |   '--- zam09.ppd
|   |-- cups5.hlp
|   |-- cupsdrv5.dll
|   |-- cupsui5.dll
|   '--- zam09.ppd
'--- WIN40
```

Nur *root* und *prtoper* dürfen in dieses Share schreiben. Dazu steht das Programm „cupsaddsmb“ zur Verfügung. CUPS liefert außerdem noch eigene DLLs-Dateien für Windows mit, die für eine Installation unter Windows notwendig ist. Sie liegt unter:

„/usr/share/cups/drivers“

Es gibt zwei weitere Alternativen.

1. Statt dem von CUPS bereitgestellten Druckertreiber kann auch der von Adobe (AdobePS) genutzt werden. Weitere Informationen siehe im Samba-Handbuch.
2. Seit Mitte 2006 gibt es eine neue Version der Windowstreiber auf der CUPS-Page verfügbar. (CUPS Driver for Windows 6).

Mit dem Befehl

```
cupsaddsmb -v zam09 ↵
```

wird für den von Samba exportierten Drucker *zam09* die entsprechende PPD-Datei und die PostScript-Bibliotheken in das *print\$*-Share kopiert.

## 3.3.2 Installation unter Windows

Ist der Samba-Server richtig aufgesetzt, lassen sich die Drucker unter Windows sehr einfach installieren.

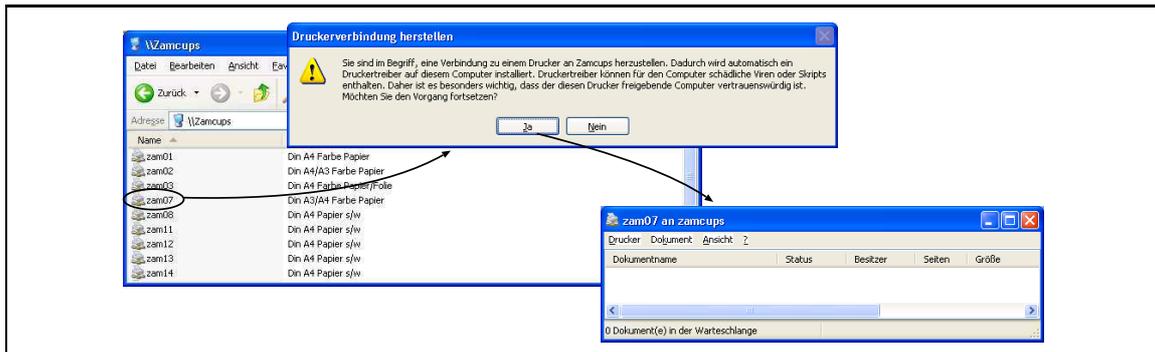


Abbildung 3.8: Samba Drucker unter Windows einrichten

Entweder man fügt einen neuen Netzwerkdrucker hinzu, oder sucht den Samba-Server (*Rechner suchen*), und läßt sich alle freigegebenen Drucker anzeigen. In Abbildung 3.8 wird gezeigt, wie die Installation abläuft. Ein Doppelklick auf einen Drucker führt zu der Meldung, daß bei dieser Aktion Treiber automatisch installiert werden. Bestätigt man das, wird der Drucker eingerichtet.

### 3.4 Alternative: CupsClient

Alle drei Methoden für Windows, die hier vorgestellt worden sind, haben gewisse Schwächen.

So muß beim Drucken mit *LPR/LPD* bzw. *IPP* für jeden Drucker ein Treiber bereitgestellt werden. Diese müssen auf der einen Seite vom Benutzer ausgewählt werden, auf der anderen Seite vom Administrator gepflegt werden.

Nutzt man Samba, existiert dieses Problem nicht. Dafür gibt es andere Schwierigkeiten. So kann es passieren, daß das Eigenschaftsfenster eines Drucker immer wieder abstürzt. Es gibt dann keine Möglichkeiten, irgendwas für das Gerät einzustellen.

Ein anderer Fehler tritt auf, wenn in einem (Word)Dokument von Landscape auf Portrait und wieder auf Landscape gewechselt wird. Der letzte Wechsel geht verloren.

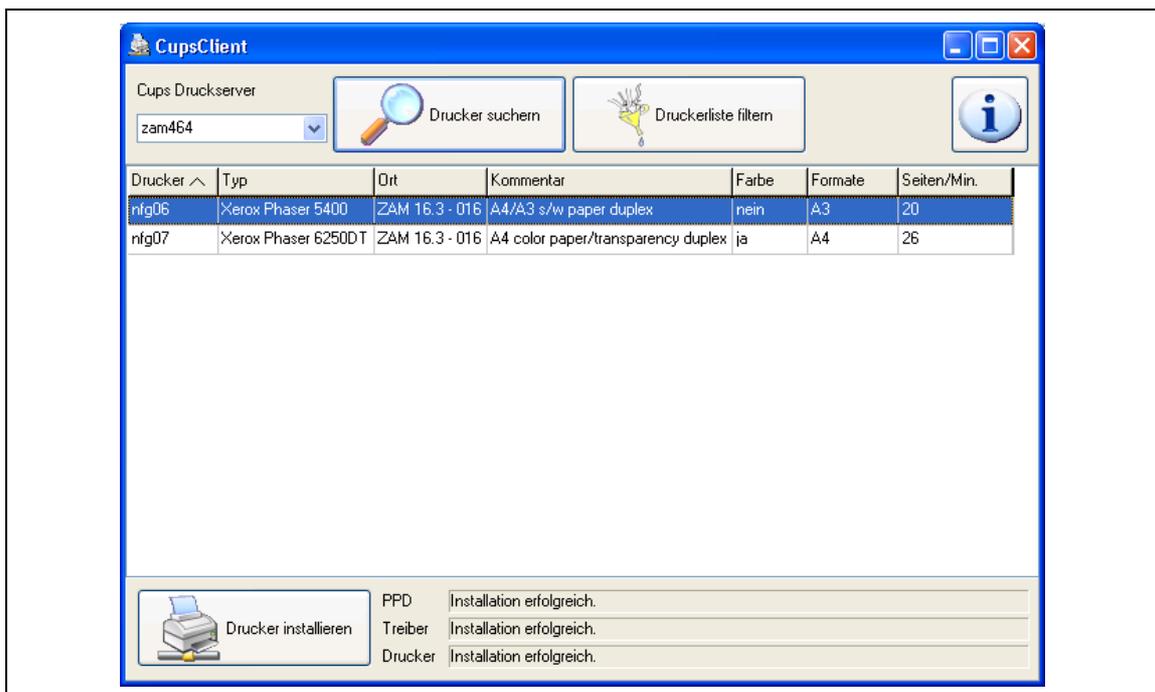


Abbildung 3.9: Drucker installieren mit *CupsClient*

**Hinweis:**

Diese Fehler sind im Zusammenspiel folgender Komponenten aufgetreten:

- CUPS 1.1.23
- Windows XP SP2
- CUPS Driver for Windows 5

Das Programm „CupsClient“ realisiert einen optimierten Weg für eine IPP-Druckerinstallation. Es kommuniziert direkt mit dem Server, und holt sich dort alle Informationen, die es benötigt. Das Besondere ist, auch der Treiber eines Druckers wird bei der Installation vom CUPS-Server heruntergeladen. Somit wird zumindest diese Lücke der IPP-Client-Implementierung von Microsoft geschlossen. Nur wenn sich der Treiber auf dem Server ändert, muß der Drucker lokal erst deinstalliert werden.

Die Software wird im Internet auf der Seite „<http://sourceforge.net/projects/cupsclient>“ angeboten, ist *Open Source* und steht unter der *GPL*.

CupsClient ist nicht ganz ausgereift, deswegen kann es zu verschiedenen Problemen kommen:

- Einige Firewalls blockieren die Kommunikation, ohne es zu melden.
- CupsClient basiert teilweise auf *CygWin*<sup>1</sup>. Ist auf einem Client eine andere *CygWin*-Version installiert als die zum *CupsClient* gehörende, kann es zu Fehlern führen.
- Auf einem englischen Windows kann es zu Fehlern bei der Installation kommen.

---

<sup>1</sup>UNIX-Umgebung unter Windows



# Kapitel 4

## Tipps rund um CUPS

### 4.1 FTP-Backend für CUPS

Im Abschnitt 2.5 auf Seite 15 wurde die Funktion der *Backends* von CUPS erklärt: Sie realisieren die Kommunikation mit den Druckern. Hier wird jetzt beschrieben, wie man ein eigenes Backend erstellt.

Einige Drucker – z.B. Poster-Plotter – werden über einem sogenannten RIP<sup>1</sup>-Rechner angesteuert. Um den Job vorzubereiten und zu drucken, ist ein Operator notwendig.

Ein möglicher Zugang zu diesem Rechner ist mittels FTP<sup>2</sup>. Da dafür kein Backend in CUPS existiert, ist hier ein Beispiel-Listing abgebildet, an dem exemplarisch der Aufbau eines selbstgeschriebenen Backends dargestellt wird.

#### Listing des *ftp*-Backends:

```
#!/usr/bin/perl -w
# CUPS FTP Backend
# The arguments passed to this script by cups are:
# 222 | zdv124 | TestJob.pdf | 1 | | /var/spool/cups/d00222-001
# $ARGV[0] - print job number
# $ARGV[1] - username
# $ARGV[2] - print job name
# $ARGV[3] - page count i think
# $ARGV[4] - ??? it is always blank
# $ARGV[5] - temporary print job location
use strict;
use vars qw();
use Net::FTP;

# prints out if no argument is given (to identify itself):
#      network ftp "Unknown" "File Transfer Protokoll (FTP)"
if ( scalar(@ARGV) == 0 ) {
    print "network ftp \"Unknown\" \"File Transfer Protokoll (FTP)\"\n";
    exit;
}
undef $/;

# log error messages
open(ERRLOG, ">>/var/log/cups/ftpperr") or die "failed to open error log: $!\n";
```

---

<sup>1</sup>Raster Image Processor

<sup>2</sup>File Transfere Protocoll

```

# get job/destination info
my $device=$ENV{'DEVICE_URI'};
$device =~ /ftp:\\/\\/(.+)\\/(.+)\//;
my $host=$1;
my $queue=$2;
my $user=$ARGV[1];
$user =~ s/\W/_/g;
my $title=$ARGV[2];
$title =~ s/\W/_/g;

# prepare job
my $local_file;
if ( $#ARGV < 5 ){
    $local_file="/tmp/fprinjob-$queue";
    open(PRINT,">$local_file")
        or die "failed to open temporary job file $local_file: $!\n";
    while (<STDIN>) {
        print PRINT ;
    }
    close(PRINT) ;
} else {
    $local_file=$ARGV[5];
}

# prepare ftp
my $ftp="undefined";
my $ftp_user;
my $ftp_passwd;
my $ftp_dir;
my $remote_file=$user."@".$queue."(".$ARGV[0].")-".$title.".ps";

# FTP-specification for destination "zam124"
if ( $host =~ /^zam124/ ) {
    $ftp_user = "prtoper";
    $ftp_passwd = "XXXXXX";
    $ftp_dir = "";
} else {
    undef $ftp;
}

# FTP-transmission
if (defined($ftp)) {
    my $rc;
    $ftp = Net::FTP->new($host, Timeout => 60, Passive => 0);
    $rc = $ftp->login($ftp_user,$ftp_passwd);
    $ftp->debug(10);
    $rc = $ftp->cwd($ftp_dir);
    $rc = $ftp->binary();
    $rc = $ftp->put($local_file,$remote_file);
    $rc = $ftp->quit();
} else {
    print ERRLOG "ERROR: Host $host not known!\n";
}
close(ERRLOG);
unlink $local_file if ( $#ARGV < 5 );
# END

```

## 4.2 variables Backend – Tea4Cups

Das *Common Unix Printing System* ist zwar modular aufgebaut, trotzdem gibt es einige Punkte, an denen es aufwendig wird, seine eigene Anforderungen zu realisieren. So ist es manchmal erwünscht, den Druckjob zu manipulieren, bevor er zum eigentlichen Drucker geschickt wird. Zur Erinnerung: „Eine Anforderung an das IPP ist es, interne Druckeranweisungen in einem Job zu überschreiben.“

Deswegen ist die sicherste Methode eine Datei auf ihrem Weg zum Drucker zu manipulieren, ohne daß CUPS noch eingreift und Änderungen vornehmen kann, wenn ein eigenes Backend eingesetzt wird. Schließlich sind die Backends das letzte Kettenglied beim CUPS-Scheduling, bevor der Job zum Drucker geschickt wird.

Nachteil ist, daß alle Backend angepaßt bzw. neu geschrieben werden müßten. Es gibt aber eine Alternative: *Tea4Cups*

Bei diesem Tool handelt es sich um ein ganz normales Backend. Nur schickt das *Tea4Cups* den Job nicht selbst zum Drucker, sondern ruft seinerseits das ursprüngliche Backend selber auf. So wird der Backend-Prozeß geschachtelt, und in der äußeren Schicht – in dem *Tea4Cups* – kann der Job noch beliebig manipuliert werden.

Der Name leitet sich ab von *tee for CUPS*. „tee“ ist ein UNIX-Befehl, der von Standard-Eingabe nach Standard-Ausgabe kopiert.

### 4.2.1 Installation von Tea4Cups

Die Homepage von *Tea4Cups* lautet:

<http://www.pykota.com/software/tea4cups>

Die Software steht zwar unter der GNU GPL<sup>3</sup>, aber die Firma *PYKOTA* verlangt für einen Account auf ihrem Server einen einmaligen Obolus von 25 USD/EUR. Mit diesen Zugangsdaten kann man im Download-Bereich die letzte „stable“-Version von *Tea4Cups* runterladen.

Die Software ist auch ohne Zugangsdaten erhältlich. Mittels `subversion` kann die aktuelle „Development“-Version heruntergeladen werden.

Vorgehensweise:

- `cd /tmp ↵`
- `svn co svn://svn.librelogiciel.com/tea4cups/trunk tea4cups ↵`
- `cd tea4cups ↵`

Folgende Dateien befinden sich u.a. in dem „tea4cups“-Verzeichnis:

NEWS:	enthält aktuelle Informationen über die letzten Änderungen
README:	beschreibt die Installation und die Konfiguration
tea4cups:	das Backend: <b>ins Backend-Verzeichnis kopieren!</b>
tea4cups.conf:	die Konfigurationsdatei: <b>nach „/etc/cups“ kopieren!</b>

Der CUPS-Dienst muß einmal neu gestartet werden, um das Backend *Tea4Cups* dem Daemon bekannt zu machen.

### Einsatz des Backends

Um *Tea4Cups* einzusetzen, muß das Backend für den Drucker angepaßt werden. Auf Seite 10 wird der Drucker „zam09“ mit der Option

```
lpadmin -p zam09 ... -v lpd://zamprt1/PS ↵
```

installiert. Soll nun das Backend *Tea4Cups* eingesetzt werden, muß der „lpadmin“-Befehl wie

<sup>3</sup>GNU General Public License

folgt angewendet werden:

```
lpadmin -p zam09 -v tea4cups://lpd://zamprt1/PS ↵
```

## 4.2.2 Konfiguration von *Tea4Cups*

Jedesmal, wenn das Backend *Tea4Cups* aufgerufen wird, wertet es seine Konfigurationsdatei „*tea4cups.conf*“ aus.

Auf drei Arten können Kommandos ausgeführt werden:

1. Filterprogramm (filter):  
Programm zum Manipulieren der Datei
2. Aktion vor dem Senden (prehook):  
Eine oder mehrere Aktionen werden unmittelbar vor dem Senden des Jobs ausgeführt. Ein Rückgabewert von „-1“ löscht den Job.
3. Aktion nach dem Senden (posthook):  
Eine oder mehrere Aktionen werden nach dem Senden des Jobs ausgeführt, es sei denn, er ist gelöscht worden.

Diese Einstellungen können für jeden Drucker separat vorgenommen werden.

Für die *pre-* und *posthooks* stehen spezielle Variablen zur Verfügung, die in den Skripten genutzt werden können (siehe Tabelle 4.1).

TEAPRINTERNAME	Druckername
TEADIRECTORY	Tea4CUPS Ausgabeverzeichnis
TEADATAFILE	Tea4CUPS-Datei (in \$TEADIRECTORY)
TEAJOBSize	Dateigröße in Bytes
TEAMD5SUM	MD5 checksum der Jobdatei
TEACLIENThOST	Hostname des Client
TEAJOBID	Job-ID
TEAUSeRNAME	Username des Senders
TEATITLe	Job-Titel
TEACOPieS	Anzahl Kopien
TEAOPTioNS	spezielle Optionen für den Job
TEAIInPUtFILE	Datendatei, leer wenn von Standardeingabe
TEABILLING	Job's billing code (lp -o job-billing=SomeCode file.ps)
TEACONTROlFILE	Control-Datei (normalerweise /var/spool/cups/c?????)
TEASTATUS	Original CUPS backend's exit code: ist nur in posthooks verfügbar.

**Tabelle 4.1:** Variablen von *Tea4Cups*

Eine Beispielkonfiguration ist in Abbildung 4.1 zu sehen.

```
[global]
prehook_create-banner : /opt/FZJ/bin/create-banner
posthook_logging : echo ">>$TEAJOBID<< ('date +\"%D %R:%S\"') :
  ${TEAUSeRNAME}@${TEACLIENThOST}, \"\$TEATITLe\", SIZE=$TEAJOBSize,
  $TEACOPieS copies, backend >${DEVICE_URI}< exit status
  >$TEASTATUS< " >>/var/log/cups/$TEAPRINTERNAME.log
[zam09]
keepfiles : yes
```

**Abbildung 4.1:** Beispiel für die Datei *tea4cups.conf*

Es gibt zwei Abschnitte, eine generelle Sektion „global“ und eine speziell für den Drucker *zam09*.

Für alle Drucker sind ein „prehook“ und ein „posthook“ definiert. Es könnten auch mehrere angegeben werden. Sie werden in alphabetischer Reihenfolge abgearbeitet.

In „prehook\_create-banner“ ist ein Shell-Skript angegeben, welches für jeden Job ein Deckblatt erstellt und hinzufügt.

Das „posthook\_logging“ erstellt zu jedem erfolgreichen Job einen Eintrag in einer für jeden Drucker separaten Log-Datei.

Beim Drucker „zam09“ werden zusätzlich alle Dateien behalten. Das kann z.B. für Fehlersuche genutzt werden.

Weitere Informationen zur Konfiguration und Nutzung von *Tea4Cups* sind dem „README“ und der Inline-Dokumentation der „tea4cups.conf“-Datei zu entnehmen.

### 4.3 Tipps zur Erstellung eines Deckblattes

In Kapitel 2.6.1 auf Seite 16 wird erklärt, wie man ein Deckblatt für CUPS erstellt. Hier in diesem Abschnitt wird eine Technik beschrieben, um die richtigen PostScript-Befehle zu erhalten, wie sie für die Fachauswahl eines HP 8500 notwendig sind (siehe Abbildung 2.13 auf Seite 17).

#### 1. Voraussetzung:

- PC mit aktuellem Windows System (XP)
- Der Drucker ist PostScript-fähig.
- Die Treiber-CD des Druckers oder von der Webseite des Herstellers den Treiber runterladen  
Wichtig ist, daß es sich um einen PostScript-Treiber handelt!  
Einige Hersteller verpacken den Treiber in einem Installations-Programm.

#### 2. Drucker unter Windows installieren:

Über das Druckermenü einen neuen lokalen Drucker installieren, als Schnittstelle „Datei/File“ angeben und den oben erwähnten Treiber auswählen.

#### 3. Druckeroptionen aktivieren:

(Drucker -> rechte Maustaste -> Eigenschaften -> Erweiterte Optionen)

Bei vielen Druckern muß angegeben werden, welches optionale Zubehör installiert ist. Erst dann können die entsprechenden Eigenschaften ausgewählt werden.

#### 4. Eigenschaften auswählen und Testdrucke erstellen:

Jedes Eingabe-Fach einmal auswählen und einen Testdruck erstellen.

#### 5. Dateien vergleichen:

Bei dem Vergleich findet man auch die Stelle, die für die Fachauswahl zuständig ist. Damit ist der erste Schritt geschafft.

#### 6. Deckblatt erstellen

- aus einem bestehenden Deckblatt:  
Hat man schon funktionierende Deckblätter, dann kann man Glück haben und die Informationen aus dem letzten Schritt einfach in eines dieser Vorhandenen einbauen. Funktioniert besonders dann, wenn man schon Drucker des selben Herstellers im Betrieb hat.
- komplett neues Deckblatt erstellen:  
Die Datei der Testseite aus dem letzten Schritt nehmen und möglichst viel daraus entfernen, bis nur noch die gewünschte Eigenschaft aktiv ist. Dann den Seiteninhalt löschen, und die eigentliche Deckblatt-Seite einfügen.

Der letzte Schritt ist der Entscheidende, aber auch der Aufwendigste. Zum einen benötigt man einen PostScript-Interpreter, z.B. *ghostscript*, der testet, ob das PostScript in der Datei noch korrekt ist:

- keine Fehler
- Stack ist leer

- Deckblatt wird angezeigt
- ein *showpage* wird erreicht

Außerdem muß eine Möglichkeit bestehen, diese Datei direkt zum Drucker zu schicken, ohne daß ein Printing-System noch irgendeine Manipulation vornimmt. Die *lpr/lp*-Kommandos besitzen die Option „-o raw“. Hintergrund für diese Option ist, daß einige Anwendungen ihren eigenen Treiber zum Drucken nutzen. In diesen Fällen soll CUPS nicht hingehen und den Job konvertieren bzw. interne Druckeranweisungen überschreiben.

Trotz alledem ist der Weg sehr aufwendig, und es wäre wünschenswert, wenn in es in zukünftigen Versionen von CUPS möglich wäre, dem Deckblatt ein Fach zuzuordnen.

# Literaturverzeichnis

- [L1] CUPS Software Administrators Manual  
*Easy Software Products (2004)*, (<http://www.cups.org/doc-1.1/sam.html>)
- [L2] CUPS Software Users Manual  
*Easy Software Products (2004)*, (<http://www.cups.org/doc-1.1/sum.html>)
- [L3] The Official Samba-3 HOWTO and Reference Guide  
*Jelmer R. Vernooij, John H. Terpstra, Gerald (Jerry) Carter*  
(<http://us1.samba.org/samba/docs/man/Samba-HOWTO-Collection>)
- [L4] TKI197: Zentral unterstützte Ausgabe von Graphik- und Textdateien  
*Stephan Graf, Herbert Schumacher (2002)*
- [L5] RFC2656: Internet Printing Protocol/1.0: Encoding and Transport  
(1999)
- [L6] RFC2910: Internet Printing Protocol/1.1: Encoding and Transport  
(2000)

