

Printing

Jason Healy, Director of Networks and Systems

Last Updated Mar 18, 2008

Contents

1	Printing	5
1.1	Introduction	5
1.2	Printing Overview	6
1.2.1	Printer Hardware	6
1.2.2	Client Machines	6
1.2.3	Older Clients	7
1.3	DNS Service Discovery (Bonjour)	7
1.3.1	Publishing a Service Zone	8
1.3.2	Publishing Service Records	8
1.3.3	Final Notes	10
1.4	CUPS Configuration	10
1.5	DHCP Configuration	10
1.6	TFTP Configuration of Printers	11
1.7	Our Helper Script	12

Chapter 1

Printing

Last updated 2008/03/18

1.1 Introduction

Printing is a service that is both popular and complex. Users want to immediately connect to printers to produce their work (especially students with papers that are overdue), but don't want to understand the complexity of drivers, PPD files, installed options, and spooling.

Over the past few years at Suffield, we've gone through several methods of printing. Very briefly:

- When we had an AppleTalk network, all printers were advertised via AppleTalk. Users with Macintosh computers simply opened the Chooser and selected the printer (by name) that they wished to use. Driver selection was automatic, or the default "generic" driver sufficed.
- As printers began to move to IP, centralized management of printers became easier, but users could not easily discover printers. We tried several print servers based on CUPS (the Common UNIX Print System) and IPP (Internet Printing Protocol). The protocol did advertise printers, but clients had to be configured to "listen" to the announcements.
- Centralized queue management was put in place to prevent printing abuses. However, the queues frequently stopped due to "errors" at the printer (such as paper jams and low toner), which required manual restarts of the queue. Users became frustrated because they could fix the underlying issue (the paper jam), but not restart the queue themselves.

- We reverted to a pure "every man for himself" system, where clients printed directly to printers. Discovery was not automatic, though, so we ended up creating a small installer program for users that would properly configure their machine with all of the printers on campus.

This leads to our current situation. Starting with Mac OS X 10.4 (Tiger), it's now possible to advertise services via a managed DNS domain. This uses the Bonjour (formerly Rendezvous) technology, but through a centralized system rather than peer-to-peer broadcasts. Clients can now auto-discover available printers and print directly to them.

This document describes the various pieces of the network puzzle that make up our printing services. It also describes a helper script that we wrote to generate the various configuration files necessary to keep printing running.

1.2 Printing Overview

Below we provide a brief overview of how all the parts of our printing system work together. The various services are discussed in more detail in the following sections.

1.2.1 Printer Hardware

When a printer is connected to the network and powered up, it asks for an IP address via **DHCP**. The DHCP server provides the printer with basic information (it's name, address, DNS servers, *etc.*). The DHCP server also provides the name of a **TFTP** configuration file. The printer fetches this file from our TFTP server, and uses it to set more advanced configuration options (available protocols, ACLs, SNMP settings, *etc.*).

Once configuration is complete, the printer goes online and is ready to accept jobs directly from users.

1.2.2 Client Machines

When a client machine connects to our network, it should notice our Service Discovery Browse Domain (via **Bonjour**/Rendezvous/Zeroconf). This is a special DNS domain that lists services (web, printing, *etc.*) on our network and the hosts that provide them.

When a user wishes to add a printer to their system, they simply use the built-in tools under Mac OS X to browse from the list of advertised printers. The make

and model of printer is published, making driver selection automatic for clients that have the proper PPD files.

Once the printer is installed, the client may print directly to the printer without further configuration. Additionally, the configured printer remains for future use.

1.2.3 Older Clients

Clients that don't support domain-wide Service Discovery (Mac OS X 10.3 and earlier) can fall back to running an installer provided by us. This installer uses CUPS command-line programs to manually add queues for all the known printers on campus. This method is less flexible than auto-discovery, but it is relatively quick (less than 1 minute), and easy (no user interaction required).

Once the queues are installed, the user simply selects the one they wish to use, and they can print directly to the printer. Installation does not need to happen again unless the network addresses of the printers change.

1.3 DNS Service Discovery (Bonjour)

To automatically advertise printers to clients, we use DNS-based service discovery (DNS-SD). Apple refers to this technology under their trade name **Bonjour** (formerly Rendezvous). While DNS-SD can be used for more than just printing, we'll focus on its printing uses in this document.

For more information on DNS-SD, please visit their web site:

<http://www.dns-sd.org/>

We use what they describe as "Wide-area DNS-SD", which is a fixed form of service discovery published via traditional DNS. Another form does exist, where clients may multicast the services they provide, but we're interested in a centrally-controlled publication of services, since we only wish to advertise services provided by the school.

In a nutshell, two things require setup in order to use DNS-SD:

1. A DNS zone for the publication of services must be created and advertised
2. Services offered by hosts must be added to this zone

1.3.1 Publishing a Service Zone

First, we create a zone to publish our service announcements in. This does not have to be a separate zone; you can use your top-level domain if you wish. However, we publish in a separate zone because our primary DNS server can't properly edit DNS-SD records (plain vanilla BIND9 works fine, so we use that server to host our DNS-SD domain).

We have called our domain `zeroconf.suffieldacademy.org`, and we host it on a BIND9 server. In order to advertise this as our SD domain, we need a few glue records in our primary domain:

```
b._dns-sd._udp.suffieldacademy.org. IN PTR zeroconf.suffieldacademy.org.  
lb._dns-sd._udp.suffieldacademy.org. IN PTR zeroconf.suffieldacademy.org.
```

These records advertise the "browse" and "legacy" browse domains in our default domain, and refer clients to the "zeroconf" domain to actually perform discovery. If you wanted, you could refer to any other domain (including the top-level domain you're already using).

Once these changes are published, you have created and advertised the domain, and you're ready to publish service records.

1.3.2 Publishing Service Records

Once you have the SD domain created, you need to create several records for each service you'd like to offer. In general, three records are required to advertise a single service. We'll discuss each one and give an example below.

Let's suppose we want to create a service offering for a printer. The name of the service should be "Faculty Room Printer" (this is a "friendly" name shown to users). When users select this service, they should print to a printer called "hplj4100-facroom.suffieldacademy.org" using IPP as the printing protocol.

The Advertisement Record

The first record links the service (IPP) to a particular offering ("Faculty Room Printer"). Clients can then ask for a service (IPP) and get an enumeration of all the hosts offering that service.

A sample record would look like this (BIND9 configuration file syntax):

```
._ipp._tcp IN PTR Faculty\032Room\032Printer._ipp._tcp
```

Note the use of escapes to convert non-alphanumeric characters into their decimal equivalent (32 is the space char in ASCII). You'll need to do this for all characters that are not valid in a DNS hostname.

This record literally says, "anyone looking for TCP/IPP service should check out the Faculty Room Printer TCP/IPP service offering". DNS automatically allows for multiple pointers, so if there are multiple offerings each one gets its own record.

The Service Record

The second record links the offering ("Faculty Room Printer") with a particular DNS hostname ("hplj4100-facroom.suffieldacademy.org"), using a standard DNS SRV record. This allows you to create "friendly" names for offerings and link them to actual DNS hostnames behind the scenes.

Here is a sample record, broken up onto two lines:

```
Faculty\032Room\032Printer._ipp._tcp IN SRV 0 0 631
hplj4100-facroom.suffieldacademy.org.
```

This time, we're associating a friendly name on the left with a specific DNS hostname and port number on the right.

The Options Record

The last record associates a DNS TXT record with the offering ("Faculty Room Printer"). This record is used to store additional information about the offering (*e.g.*, printer make and model, queue name, and options). In this way, clients can auto-configure themselves to use the offering.

The configuration is stored as a series of key=value pairs, lumped together in a single TXT record. Here is a sample record in BIND9 format:

```
Faculty\032Room\032Printer._ipp._tcp IN TXT (
"txtvers=1"
"Color=F"
"Duplex=T"
"note=Faculty Room, next to copier"
"product=(HP LaserJet 4100 Series )"
"ty=HP LaserJet 4100n"
)
```

The specific key/value pairs are described in more detail on the DNS-SD site. The two most important (for the end user) are the "note" key, which provides

a description to the user, and the "product" key, which helps the client find a suitable PPD to autoconfigure the printer.

1.3.3 Final Notes

Those three records must exist for every service offering. Note that if a printer supports multiple protocols (*e.g.*, IPP and LPD), you'll need a set of three records *for each protocol*. In other words, you'd need three records under the `_ipp._tcp` domain (for IPP), and three more under the `_printer._tcp` domain (for LPD).

1.4 CUPS Configuration

For computers that don't yet support DNS service discovery, we fall back to providing an installer that creates the queues for them. All Mac OS X machines (since 10.2) use CUPS as their underlying print client, and we can configure queues for CUPS directly from the command line.

We create a simple shell script that adds print queues using the `lpadmin` command:

```
lpadmin -p "hplj4100-facroom.suffieldacademy.org" -E \  
-v "ipp://hplj4100-facroom.suffieldacademy.org/ipp/" \  
-L "Faculty Room, next to copier" \  
-D "Faculty Room Printer"
```

Note that it specifies the host, URL (IPP in this case), location, and "friendly" name. We simply run this for every printer available on campus.

We place this shell script inside an empty Mac OS X installer package. The package doesn't install any files, but instead runs the script as a "postinstall" and "postupgrade" step. This allows the user to have a simple familiar interface (package installation) which runs a shell script.

1.5 DHCP Configuration

To make our life easier, we use DHCP (and TFTP; see below) to auto-configure printers. This doesn't have anything to do with auto-discovery of printers by clients; it's just to make printer management easier from our end.

This is not a document on DHCP; for that, we recommend you check out [Our documentation on running DHCP](#). Briefly, you'll need to do a few things:

1. First, if you're going to use TFTP (see next section) with HP printers (what we use), you'll need to enable a few options to make this possible. In the global section of your DHCP config, add options 144 and 150 to your server:

```
option jd-tftp-config code 144 = string;
option jd-tftp-server code 150 = string;
```

As you can see, 144 defines the TFTP configuration file path, and 150 defines the TFTP server to use.

2. Next, define the TFTP server (if you don't, the printer will default to using the DHCP server):

```
option jd-tftp-server "tftp.suffieldacademy.org";
```

3. Finally, create a host stanza for every printer, and assign the printer a name and TFTP config file:

```
host hplj4100-facroom.suffieldacademy.org {
    hardware ethernet de:ad:be:ef:f0:0d;
    ddns-hostname "hplj4100-facroom.suffieldacademy.org";
    option jd-tftp-config "/private/tftpboot/printers/deadbbeff00d";
}
```

Note that we store the configuration by MAC address. This parameter must be kept relatively short, as some printers won't accept a filename that is too long.

Once you've finished setting up DHCP and TFTP, you can power up the printers and they should be automatically configured.

1.6 TFTP Configuration of Printers

The printers we use (HP LaserJets) have the ability to take configuration options via TFTP at boot time. By specifying the path to a TFTP file in the DHCP options for a printer (see previous section), the printer will get this path when it boots and try to read that configuration file.

This process doesn't have anything to do with making the client experience better; it's just to make managing printers a little easier. You can skip this process and still have auto-discovered printers using only DNS-SD.

The configuration file is in the same format as a telnet-based config; any command you can issue via telnet should also work when placed verbatim inside a

file. We suggest telnetting to your printer and using the "export" command to produce a file that you can then customize.

Once you have the configuration file the way you want it, simply make it available on a TFTP server. TFTP servers are available for all UNIX-based platforms, and Mac OS X comes with one built-in. Just enable it with:

```
sudo launchctl load -w /System/Library/LaunchDaemons/tftpd.plist
```

To make the configuration files available, just put them in `/private/tftpboot` (or wherever your TFTP server path is). You can even place the configuration files in their own directory to keep them all together.

Once that's done, just provide the path to the config in the DHCP host stanza. The next time the printer boots, it should look for the configuration file option, try to fetch the file via TFTP, and load its contents. Check your syslog (or configuration page) for errors in this process.

1.7 Our Helper Script

To assist with all of these different configuration files, we've created a unified helper script. The script takes a single description file as its input, and produces configuration file snippets for all the services mentioned previously (DNS, DHCP, TFTP, and CUPS).

The script is well-documented, including a sample configuration file. Please [download the script](#) from our website and use "perldoc" on it to read the built-in documentation:

```
perldoc exporter
```

To use the script, simply list the services you want to generate, followed by the input file and an output directory. For example, to generate all service files, you would do the following:

```
./exporter tftp dhcp dns cups printers.dat output_dir
```

The output directory will now contain a partial DNS zone file, DHCP host stanzas, a CUPS installer script, and a directory of TFTP configuration files.

The script has two dependencies: a directory of template files used for creating the TFTP configurations, and a directory of PPD files that can be used by the CUPS installer.

The TFTP template files are named by printer model, or by exact hostname. If you like, you can [browse our template directory](#) online.

The CUPS PPD files are simply the PPDs provided with the printer, or that we have used from machines that work. You should download or use PPDs that are known to work with your printers.