Getting a USB receipt printer working on Windows

mike42.me /blog/2015-04-getting-a-usb-receipt-printer-working-on-windows

mike

Note:*This post is a Windows adaptation of an earlier post, Getting a USB receipt printer working on Linux, mainly in response to these questions.*

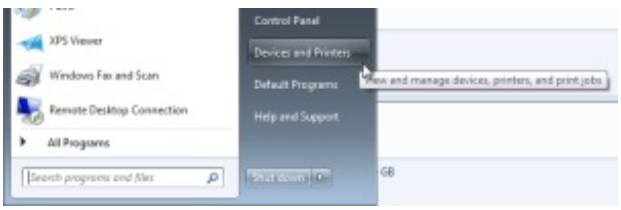
In this post, I'll step through how to get a USB thermal receipt printer appearing on Windows. The aim of this is to be able to send raw text to the printer, so that we can point a driver such as escpos-php at it. The printer tested here is once again this Epson TM-T20:

The directions below are for Windows 7, so your mileage may vary if you are on an older or newer version.

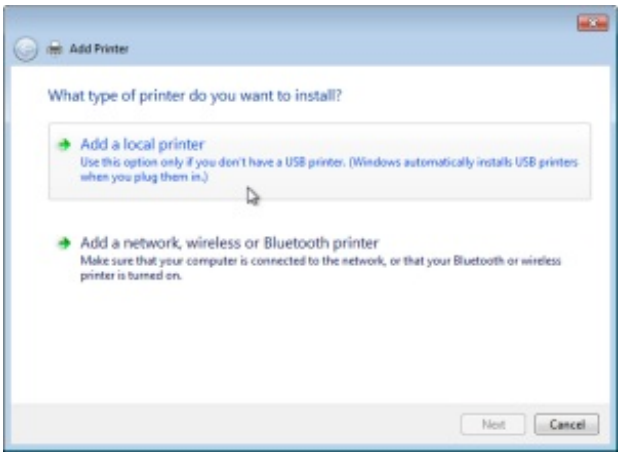
If you have issues following these steps, make sure you can locate your printer in Device Manager, and that it has "USB Print Support".

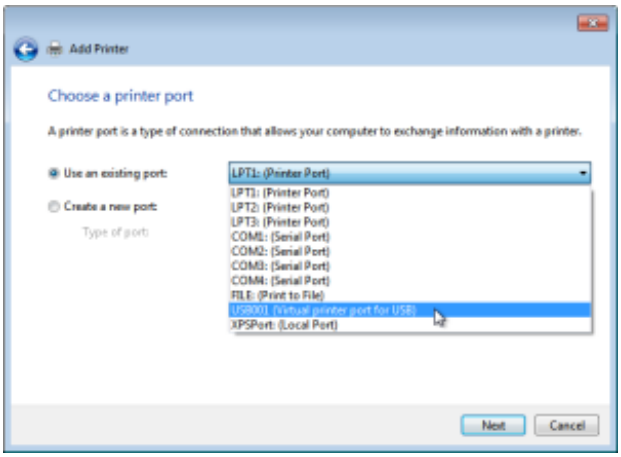
Add the printer

Find **Devices and Printers** and click **Add a Printer**.

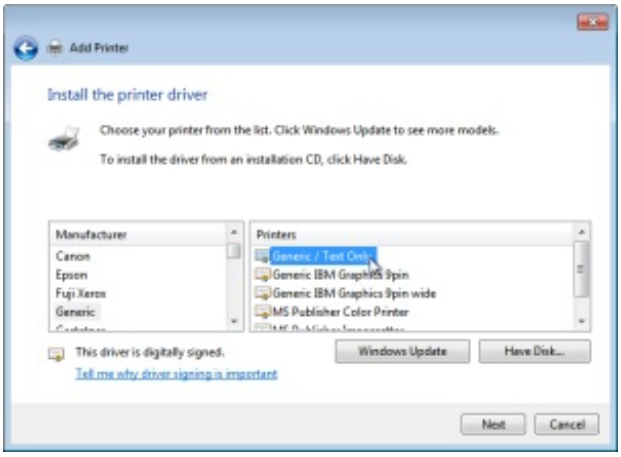


Add it as a **Local printer**, using the USB virtual port, probably **USB0001**:

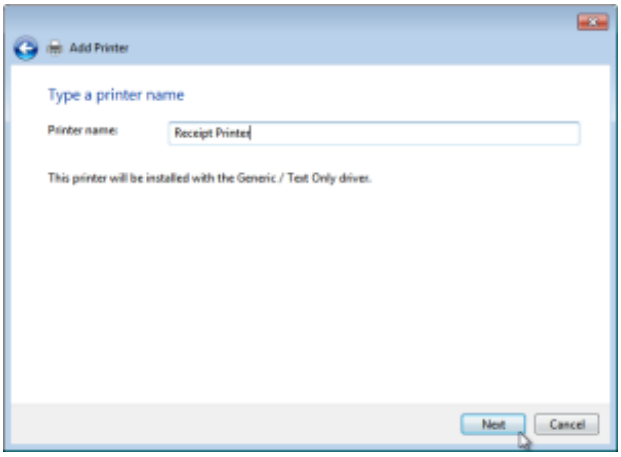


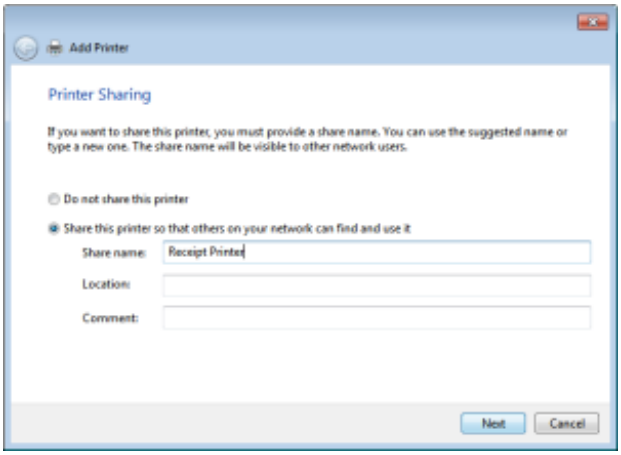


Use the **Generic / Text Only** driver.

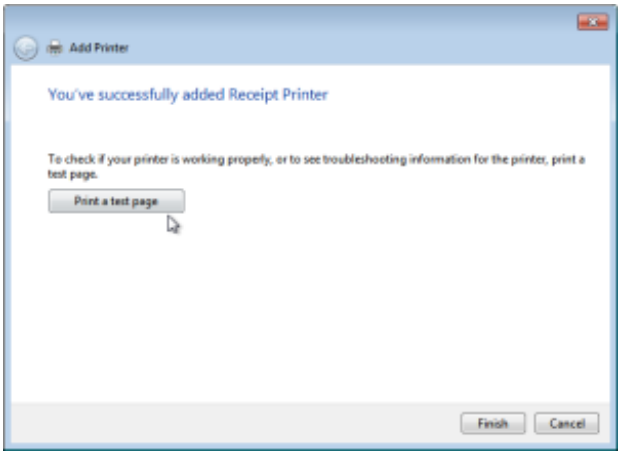


Name the printer whatever you like, and then share it under the same name:



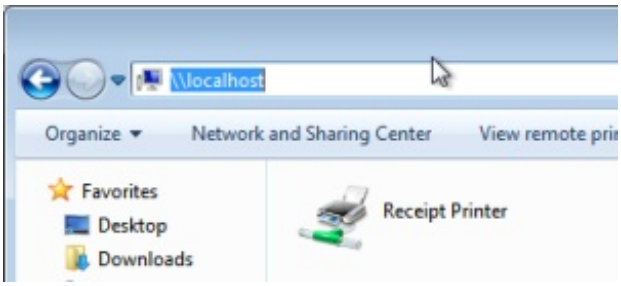


At this point, it should pop up in the window in the background, and also prompt you to **Print a test page**.



The test print is plain-text, and depending on your printer, will look something like this:

Finally, you need to verify that your printer can be accessed locally, by typing `\\localhost` into Windows Explorer. If all goes to plan, you will see the new printer there too:



Run a command-line test print

We now know that your printer is working, and can be accessed via its share name (even locally).

Test printing from the command-line. Fire up `cmd.exe` and try to send it some text to verify that it's working:

```
echo "Hello World" > testfile
print /D:"\\%COMPUTERNAME%\Receipt Printer"
testfile
del testfile
```

Printing something useful

This is where you start to see real results. Receipt printers are not just for printing plain-text. Many of them support a standard called ESC/POS, which contains formatting commands.

The snippet below, from this earlier post, generates some basic ESC/POS commands.

Install PHP if you don't have it already, and call the below code `foo.php`:

```
<?php
/* ASCII constants */
const ESC = "\x1b";
const GS="\x1d";
const NUL="\x00";

/* Output an example receipt */
echo ESC."@"; // Reset to defaults
echo ESC."E".chr(1); // Bold
echo "FOO CORP Ltd.\n"; // Company
echo ESC."E".chr(0); // Not Bold
echo ESC."d".chr(1); // Blank line
echo "Receipt for whatever\n"; // Print text
echo ESC."d".chr(4); // 4 Blank lines

/* Bar-code at the end */
echo ESC."a".chr(1); // Centered printing
echo GS."k".chr(4)."987654321".NUL; // Print
barcode
echo ESC."d".chr(1); // Blank line
echo "987654321\n"; // Print number
echo GS."V\x41".chr(3); // Cut
exit(0);
```

You would send generated commands to the printer like this:

```
php foo.php > testfile
print /D:"\\%COMPUTERNAME%\Receipt Printer"
testfile
rm testfile
```

Scaling this up

The correct ESC/POS codes are quite tricky to generate with manually, which is why I put together the escpos-php driver. You can find more information on that at:

- escpos-php on GitHub
- What is ESC/POS, and how do I use it?

A simple “Hello World” receipt to your Windows shared printer would be scripted as (call this one `foo2.php`):

```
<?php
require __DIR__ . '/autoload.php';
use Mike42\Escpos\Printer;
use Mike42\Escpos\PrintConnectors\WindowsPrintConnector;

try {
 // Enter the share name for your USB printer here
 $connector = new WindowsPrintConnector("Receipt Printer");
 $printer = new Printer($connector);

 /* Print a "Hello world" receipt" */
 $printer -> text("Hello World!\n");
 $printer -> cut();

 /* Close printer */
 $printer -> close();
} catch(Exception $e) {
 echo "Couldn't print to this printer: " . $e -> getMessage() .
"\n";
}
```

This would be sent to the printer by loading it from the web, or running the script on the command-line:

```
php
foo2.php
```

The full ESC/POS snippet with formatting, coded up with escpos-php, would look like this (call this one `foo3.php`):

```
<?php
require __DIR__ . '/autoload.php';
use Mike42\Escpos\Printer;
use Mike42\Escpos\PrintConnectors\WindowsPrintConnector;
try {
 // Enter the share name for your USB printer here
 $connector = new WindowsPrintConnector("Receipt Printer");
 $printer = new Printer($connector);

 /* Print some bold text */
 $printer -> setEmphasis(true);
 $printer -> text("FOO CORP Ltd.\n");
 $printer -> setEmphasis(false);
 $printer -> feed();
 $printer -> text("Receipt for whatever\n");
 $printer -> feed(4);

 /* Bar-code at the end */
 $printer -> setJustification(Printer::JUSTIFY_CENTER);
 $printer -> barcode("987654321");

 /* Close printer */
 $printer -> close();
} catch(Exception $e) {
 echo "Couldn't print to this printer: " . $e -> getMessage() .
"\n";
}
```

And again, this could be executed by loading the page through the web, or invoking the command directly:

```
php
foo3.php
```