

BARZ_OUT™ Pro

FOR LINUX & UNIX

© 2008 by Unibar, Inc.
All Rights Reserved.

Unibar and BARZ_OUT™ Pro are trademarks of Unibar, Inc. All brand and product names used in this manual are trademarks or registered trademarks of their respective companies.

3345 Auburn Road, Suite 201
Rochester Hills, MI 48309

(248) 299-5050 (General)
(248) 299-5050 x290 (Sales)
(248) 299-5050 x291 (Tech Support)
(248) 299-5052 (Fax)

Email: unibar@unibar.com
Web: www.unibar.com

Table of contents

INTRODUCING BARZ_OUT™ PRO	3
SYSTEM REQUIREMENTS	3
INSTALLING BARZ_OUT™ PRO	3
Environment Variables	3
BARZ_OUT™ PRO OPERATION	3
WHAT'S NEW WITH VERSION 5.0	5
PostScript Option	5
UPC 2 and 5 Digit Supplements	5
Barcode Trigger Length	5
CUPS Printing System	5
GETTING STARTED	6
Sample Command Lines	6
Configuration File	6
Registration	6
License Key File	7
Display License Key (unibar.key) Information	7
DEFAULT SETTINGS	8
SETTING AVAILABLE CHOICES	8
Barcode Type	8
Height	9
Density	9
Human Readable Printing	10
Checksum	10
Rotation	10
Leading Trigger ~ (tilde)	10
Trailing Trigger (blank)	11
Printer Type	11
Bin	11
Lines Per Inch	12
Characters Per Inch	12
Next Barcode Type Definition	12
Text Fields (Bold, Italics, Underline, Point Size, Reverse Video)	13
Print Job Options: (Orientation, Line Termination Mode, Reset)	14
USING MULTIPLE TRIGGERS	16
BARCODING CONTROL CHARACTERS	16
BARZ_OUT™ PRO TIPS	16
APPENDIX A - LISTING OF PARAMETERS	18
APPENDIX B - PDF417 PARAMETERS	19
Handling of Rows, Columns and Aspect Ratio of the overall Barcode	19
APPENDIX C - EAN APPLICATION IDENTIFIERS	21
APPENDIX D – EXAMPLES OF CONFIGURATION FILES	24
APPENDIX E – INSTALLING BARZ_OUT™ AS A CUPS FILTER	25

INTRODUCING BARZ_OUT™ Pro

BARZ_OUT™ Pro is a "print filter" program that accepts printer input from any standard input device (normally a program or a file), scans the data for trigger characters and replaces the data with barcodes in accordance with the configuration file parameters. The data is then sent to the printer specified on the command line or standard out if no print destination is specified with an **Enterprise** license. Please note that with a **Printer** license, output may only be sent to configured print devices as specified in the Print Destination Table.

SYSTEM REQUIREMENTS

BARZ_OUT™ Pro requires about 200K and one megabyte disk space. Validate your Linux or Unix vendor/version number with the current support list.

Installing BARZ_OUT™ Pro

Create a directory in which to store BARZ_OUT™ Pro and do a change directory to make it your present working directory. For example:

- o `mkdir unibar`
- o `cd unibar`
- o Copy the compressed file and the install script to the *unibar* directory
- o Run the install script by typing `./installUB.sh` and follow the instructions. (installUB.log will be created)

(for more instructions on how to install from the web, go to www.unibar.com)

Environment Variables

BARZ_OUT™ Pro uses a base directory to locate the configuration files and the license key file. Normally, this directory is `"/unibar"`. The user can use any desired directory, but the BARZ_OUT™ Pro program must be informed. This is done by setting the environment string.

```
$UNIBAR=/usr/unibar
$export UNIBAR
```

The commands above can be inserted in a login script so they are automatically executed whenever the user logs on. This directory should also be added to the PATH. Otherwise, full paths must be used or a `./` must be prepended to the file names when in this directory.

BARZ_OUT™ Pro Operation

BARZ_OUT™ Pro takes three parameters: input, configuration, and output. Input is the standard input stream (*stdin*). This can be a file or data piped from another program. Configuration (such as `sample.cfg`) is a file that instructs BARZ_OUT™ Pro as to which barcode type to use, what printer type, etc. The configuration file is required. Output depends on the license that comes in two forms: **Enterprise**, for medium to large organizations with many print destinations and **Printer**, for small operations with very few printers.

With an **Enterprise** license, bz sends the barcode drawing commands inline with the other document contents to the standard output stream. This output can then be redirected to a file, sent directly to a printer, or piped to another program or a print spooler like lp.

With a **Printer** license, output cannot be redirected and bz looks for a Logical name on the command line (with the /P=LogicalName switch) that matches a print device in the Print Destination table found in the file barzout.pdt. An example of a barzout.pdt file:

```
[Print Destination Table]

PRT_DEST =
{
    Logical = defaultPrinter
    Physical = lp
    TYPE = PIPE
}

PRT_DEST =
{
    Logical = dept2Printer
    Physical = lp -d PrintQName
    TYPE = PIPE
}

PRT_DEST =
{
    Logical = LocalUNIXPrinter
    Physical = /dev/lp0
    TYPE = DEVICE
}

Default = defaultPrinter
```

The output of BARZ_OUT contains native printer commands for drawing the barcodes. As such, it is ideal for this output to be sent directly to the printer without any “interpretation” by a printer driver – which may cause undesired results. The output of BARZ_OUT should be sent to a “raw” print queue. If not, then a “raw” option should be specified. In the first print destination of the example PDT above, the Physical entry would include this as:

```
Physical = lp -o raw -d PrintQueueName
```

Please see Appendix E – Installing BARZ_OUT as a CUPS Filter for more discussion of print queues.

What's New With Version 5.0

PostScript Option

Support for the PostScript printer language has been added. BARZ_OUT will now generate PostScript output as well as scan properly formatted PostScript input. This new feature is accessed with a new Printer Type option.

UPC 2 and 5 Digit Supplements

UPC barcode processing now allows for the addition of the 2 and 5 digit supplements as add-on to the primary barcode sequence. This allows the UPC barcode and supplement barcode to be handled by a single trigger. These supplements are typically used for specifying the issue number and suggested retail pricing.

Barcode Trigger Length

The maximum barcode trigger length has been extended to 30 characters (upper and lower case). This allows for XML style begin and end trigger sequences.

CUPS Printing System

Barz_out can now be installed as a CUPS (Common Unix Printing System) filter. Having this application as part of the printing system makes the barcode conversion process automatic.

GETTING STARTED

Sample Command Lines

Note: '<' and '>' are redirect symbols and should be used on the command line as-is.

File input from a file called "sample.txt", output to a file "sample.out" with Enterprise license:

```
bz -CFGsample.cfg < sample.txt > sample.out
```

↑ ↙ ↑ ↗
Program Name Configuration File Input File Output File

Piped input from a user program called "user" with Enterprise license:

```
user | bz -CFGuser.cfg > user.out
```

Output sent directly to a printer "/dev/lp0" with Enterprise license:

```
bz -CFG textAttributes.cfg < textAttributes.txt > /dev/lp0
```

Output piped to a program "lp" (the print spooler on many UNIX systems) with Enterprise license:

```
bz -CFGsymbolTriggers.cfg < symbolTriggers.txt | lp
```

Output to a device in Print Destination Table with Printer license:

```
cat xmlTriggers.txt | bz -CFGxmlTriggers.cfg -p=dept1Printer
```

Configuration File

The configuration file is named on the command line, so a user can define several different configurations for different documents and save them with unique filenames (invoice.cfg, shiporder.cfg, etc). In the case where a user wants to use all but one setting from a configuration file, a command line override can be used. A command line override is specified by typing a line from the configuration file on the command line. For example, suppose a user had a configuration file "user.cfg". Suppose the user wants to use BARZ_OUT™ Pro as configured in "user.cfg" except that the user wants a taller barcode. The following command would generate bars a half inch tall:

```
cat sample.txt | bz -CFGsample.cfg -HEIGHT=50 | lp -d PrintQName
```

Registration

You must register BARZ_OUT™ Pro in order to get a permanent license key. Run the BARZ_OUT Pro command with a '-ir' option, and follow the instructions on the screen.

Example `bz -ir` or `./bz -ir` if the unibar directory is not in the PATH.

License Key File

BARZ_OUT™ Pro reads a license key file to determine the features it should enable. The file is named "unibar.key". The program looks for the key file in the directory that is referenced by the \$UNIBAR environment variable. BARZ_OUT comes with a Demo key that allows full use of all features of the software, but will change one character or digit in each field that it processes. The license file is an encrypted file, manually editing the file can make it unusable.

Display License Key (unibar.key) Information

At the command prompt, in the directory where BARZ_OUT™ Pro was installed, type:

```
bz -i or ./bz -i
```

The license information in the unibar.key file will be displayed to the screen, and look similar to:

```
/unibar $ ./bz -i  
The '-i' option displays this message.  
'-ir' creates a file for registration (register.txt).  
'-id' creates a file for debug (debug.txt). The '.txt' files  
can be emailed to techsupport@unibar.com
```

```
Barz_Out Pro(2D), Version 5.0.3, Copyright Unibar, Inc. 1995-2008.  
O/S: Name Linux, Release 2.6.9-34.ELsmp, Version #1 SMP Fri Feb 24 16:54:53 EST 2, Machine i686  
Serial Number: 1234567890, Node name: linux1.unibar.com  
License File: /unibar/unibar.key  
BARZ_OUT Pro (2D) (UNLIMITED printers) licensed in DEMO MODE
```

This command is also helpful for determining how many days are left before an annual key will expire or if there is a serial number mismatch.

```
Barz_Out Pro(2D), Version 5.0.3, Copyright Unibar, Inc. 1995-2008.  
O/S: Name Linux, Release 2.6.9-34.ELsmp, Version #1 SMP Fri Feb 24 16:54:53 EST 2, Machine i686  
Serial Number: 1234567890, Node name: linux1.unibar.com  
License File: /unibar/unibar.key  
BARZ_OUT Pro (UNLIMITED printers) licensed until Nov 28,2007 (13 days remaining)
```

DEFAULT SETTINGS

The settings shown below are BARZ_OUT™ Pro's built in Default settings. Each option setting and the available choices for each is listed and defined on the following pages.

OPTION SETTING DEFAULT

Barcode Type	3 of 9
Height	25
Density	High
Text Printing	None
Checksum	Disabled
Leading Trigger	~ (tilde)
Trailing Trigger	(blank)
Bin	All Bins
Lines Per Inch	6 lpi
Chars Per Inch	10 cpi

Please note that there is no default for Printer Type, it must be specified in a cfg file.

SETTING AVAILABLE CHOICES

Barcode Type

This setting selects the barcode symbology to be used when encoding the data. Each type has its own character sets, specifications etc. BARZ_OUT™ Pro adheres to all these criteria when barcoding your data. It assures that the data you barcode follows all guidelines specified by the agencies responsible for the development of the various barcode standards.

<u>Symbology</u>	<u>Config File Setting</u>	<u>Symbology</u>	<u>Config File Setting</u>
3 of 9	-3_OF_9	Ship To Location Code	-EAN_SHIPLOC
Extended 3 of 9	-EXT_3_OF_9	Ship To Postal Code within Single Postal Authority	-EAN_SHIPSPA
Codabar	-CODABAR	Plessey	-PLESSEY
Code 128	-CODE128	Postnet	-POSTNET
Interleaved 2 of 5	-2_OF_5	PDF417 †	-PDF417
EAN 8	-EAN_8	Shipping Container Code	-SCC_14
EAN 13	-EAN_13	Serial Shipping Container Code	-SSCC_18
Batch Numbers	-EAN-BATCH	UCC EAN 128	-UCC_EAN_128
Electronic Serial number for cellular phone	-EAN_CELL	UPC 2	-UPC_2
Expiration Date	-EAN-EXP-DATE	UPC 5	-UPC_5
Package Date	-EAN-PKG-DATE	UPCA	-UPC_A
Production Date	-EAN-PROD-DATE	UPC E	-UPC_E
Sell By Date	-EAN-SELL-DATE	UPC Version D	-UPC_VERS_D
Ship To Postal Code with 3-digit ISO Country Code Prefix	-EAN_SHIPISO		

† Only available with 2D option

Height

This selection controls the distance vertically that BARZ_OUT™ Pro should use to barcode your data. This height selection does not take into account the addition of human readable text. If you have selected the text printing option (below) the total height used by BARZ_OUT™ Pro will be your height selection plus one line for the printing of your text. Care should be taken to assure BARZ_OUT™ Pro has enough room to create barcode. Do not include text or graphics in the area that is to contain your barcode. The height setting is set in 1/100 th of an inch.

Example: To set the height of the barcode to 1/4"

Config File Setting: -HEIGHT=25

Density

The density option defines for BARZ_OUT™ Pro the width to use for the narrowest element in the barcode. This density setting provides the user with three options controlling the resolution printed. The HIGH setting allows for a narrow element of approximately 7.5 mils (.19mm). The medium setting is approximately 15 mils, and the low setting is approximately 23 mils. Actual dimensions vary based on the specific printer type being used. A ratio can be set, be it is recommended that the default of 2.6:1 be used. If you do not set the ratio, the default ratio will be used.

High	Highest resolution (thinnest bars)	7.5 mils
Medium	Medium resolution	15 mils
Low	Lowest resolution (wide bars)	23 mils

For more control over the density, explicitly specify the narrow bar width.

Config File Setting:

-DENSITY=50 & Above The narrow bar width in tenths of mils.

Example: 50 = 5 mils
 65 = 6.5 mils
 76 = 7.6 mils

-RATIO=2.6 This number can be 2.0 to 3.0.

UPC 2 and 5 Digit Supplements:

UPC barcodes may be specified with a 2 or 5 digit extension. Normally, these 2 and 5 digit supplements are used to indicate magazine/newspaper issue numbers and suggested retail price respectively. To include a 2 or 5 digit supplement, use the '/' character to separate the two sets of number sequences. These may be utilized as follows: LTRIGGER=<upca>

<upca>123456789012/12	- would produce a UPC barcode with issue number = 12
<upca>210987654321/51999	- would produce a UPC barcode with price = \$19.99

Where the leading 5 indicates the currency is in dollars.

Human Readable Printing

This option allows the user to specify the addition of actual text of the barcode to be included. The text, if desired, may be printed above or below the barcode. This option will print all data coded including checksum characters if present.

Config File Setting:

-TextNone

No human readable text



-TextBelow

Readable text below barcode



-TextAbove

Readable text above barcode



Checksum

Check characters are a method used to assure the integrity of the data read by barcode scanning equipment and software. Though most barcode symbologies are self checking and highly reliable, occasionally a user may desire the added security a check digit affords. When selected, BARZ_OUT™ Pro will include in the barcode data a check character, generated by an algorithm specified by the various barcode symbology. When printing human text, this check digit will be printed also. If you do not require a check character, or your data already includes one, leave this option disabled.

Config File Setting:

-NoChecksum

12345



-Checksum

('F' is the checksum character)

12345F



Rotation

Barcodes can be printed at 0, 90, 180, or 270 degrees using the -O_value switch (dash and letter O).

Config File Setting: -O_0 (for zero degrees)
-O_90 (for ninety degrees)
-O_180 (for one hundred eighty degrees)
-O_270 (for two hundred seventy degrees)

Leading Trigger ~ (tilde)

The lead trigger character(s) tell BARZ_OUT™ Pro that the data that follows is to be barcoded. The trigger will be stripped off of the data and not included in the barcode. The lead trigger also tells BARZ_OUT™ Pro where to begin the barcode on the document being printed. The trigger should be placed where the upper left hand edge of the barcode is to be printed. The '~' tilde character is the default trigger. This may be changed by the user to another character or a short string of characters. Care should be used when selecting a trigger, as to guarantee its uniqueness in the data being printed. The appearance of the trigger elsewhere may cause BARZ_OUT™ Pro to barcode data that was not intended to be coded.

Trailing Trigger (blank)

The trailing trigger indicates to BARZ_OUT™ Pro that the end of the data for the current barcode has been reached. This data will not be included in the barcode. By default a blank or end of line will cause BARZ_OUT™ Pro to terminate the current barcode. It is suggested that the user change from this default setting unless only straight ASCII will be used, as was the case in early versions. It remains the default for backward compatibility. Care should be taken when selecting a trailing trigger to assure that all data to be included in a barcode is unique from the trigger.

Config File Setting: -TTrigger=</code128>

Printer Type

Select the printer or model which is closest to your printer. Your printer manual will indicate which printer emulation modes, or compatibility options it can support.

Config File Setting:

-HPLASER	Hewlett Packard LaserJet (or any PCL Laser Printer)
-EPSONFX	EPSON FX
-EPSONLQ	EPSON LQ
-IBMPRO9	IBM Proprinter 9 Pin
-IBMPRO24	IBM Proprinter 24 Pin
-POSTSCRIPT	Adobe PostScript

Bin

Selected bin media will be barcoded. Input from other bins will not be barcoded, i.e. if envelope feeder is selected only envelopes from the envelope feeder (bin 6) will be barcoded. For HP & PCL compatible printers only.

Config File Setting:

-AllBins	All Bins
-UpperTray	Upper Tray
-ManualTray	Manual Tray
-ManualEnv	Manual Envelope
-LowerTray	Lower Tray
-EnvFeeder	Envelope Feeder

Lines Per Inch

Selects the number of lines per inch. Most printers use 6 lines per inch as their default. Only change this setting if you have selected 8 lines per inch manually on your printer's control panel, or in the application containing the data you are printing. When changing this option you must go back and adjust the height setting to assure the proper height barcodes are generated for your application. The height options are dependent on the lines per inch setting. For HP & PCL compatible printers only.

Config File Setting:

-6LPI	6 lpi (default)
-8LPI	8 lpi

Characters Per Inch

Selects the number of characters per inch your application is using to print data. This value may also be referred to as pitch. The setting of this value accurately assures BARZ_OUT™ Pro can properly place barcode(s) on your form. Use of proportional spaced fonts is not recommended as barcode placement may not be accurate. For HP & PCL compatible printers only.

Config File Setting:

-7CPI	7 cpi
-10CPI	10 cpi
-12CPI	12 cpi
-17CPI	17 cpi
-25CPI	25 cpi

Next Barcode Type Definition

Starts a new barcode definition. This barcode definition may have different trigger characters, barcode type, height, etc... This command allows you to define more than one barcode type within a single document.

Config File Setting: -NEXTBARCODE

Text Fields (Bold, Italics, Underline, Point Size, Reverse Video)

Users can setup trigger characters to define a type of text. For HP & PCL compatible printers only.

Config File Setting:

BOLD

The value of the BOLD setting can range from '7' (very dark) to '-7' (very light), including '0' (normal). There are three forms of this command:

- BOLD selects a value of '7' for the selected text.
- BOLD=*d*, selects a value of 'd' for the selected text (-7 <= d <= 7).
- BOLD_OFF=*d* after printing the selected text, the value for BOLD is set to '0' by default. The user can use BOLD_OFF to specify a different exit value.

ITALICS

-ITALICS

UNDERLINE

-UNDERLINE

POINT SIZE/PITCH

The Point Size feature is intended to accomplish fairly simple text sizing functions. The font sizing capabilities of PCL, meanwhile, are rich in options and consequently, not really simple.

A particular issue of importance here is that the PCL Height command "is ignored when selecting a fixed-space scalable font", while the PCL Pitch command is "ignored when selecting a proportionally-spaced (bitmap or scalable) font".

To keep Barz_Out Pro simple, it merely passes on the Height or Pitch command as directed in the trigger. This means that the user will need to know whether the font is fixed-space or proportionally-spaced.

In practice, it means the user may typically try the POINT_SIZE and the PITCH and see which one works.

- POINT_SIZE= The form for Point Size is "-POINT_SIZE=fff", where 'fff' is a floating point number that can range from 1 to 999.
- PITCH= The form for Pitch is "-PITCH=fff", where 'fff' is a floating point number that can range from 576 to 0.1.

REVERSE VIDEO

This command causes the following sequence to happen: the text "print model" is essentially set to white on black, the background is shaded black and then the text is printed in white.

The variations of this command revolve around how to specify the size of the black background. REV_VIDEO_POINT sets the height of the background. The length can either be a fixed size or can vary depending on the length of the string printed.

- REV_VIDEO_POINT= To set the height of the background, use REV_VIDEO_POINT=ddd. The units are 1/100 inch . Note that this does not set the point size of the text itself, only the height of the background.
- REV_VIDEO_LEN= To set a fixed length background, use REV_VIDEO_LEN=ddd. The units are 1/100 inch.
- REV_VIDEO_CPI= To set a variable length background, us REV_VIDEO_CPI=ddd. The units are 1/100 inch.

Print Job Options: (Orientation, Line Termination Mode, Reset)

The print job options should only be used if your spooler/application does not already format the print job. An example of when to use the print job options would be when a Unix/Linux print job is sent to a remote printer/queue that does not support print job formatting. Also, in testing, Unibar has found that when sending a print job from a Unix/Linux environment to a Windows environment that the print job options should be used to format the document.

Config File Setting:

Page Orientation

The "PAGE_ORIENT" option instructs Barz_Out Pro to send a Logical Page Orientation command to the printer. This is a PCL command and the PAGE_ORIENT option is useful for ***PCL type printers only***. It is **ignored** for all other printer types.

The command takes one of four parameters, as follows:

- | | |
|--------------------------------|------------------|
| -PAGE_ORIENT=PORTRAIT | Rotate page 0° |
| -PAGE_ORIENT=LANDSCAPE | Rotate page 90° |
| -PAGE_ORIENT=REVERSE_PORTRAIT | Rotate page 180° |
| -PAGE_ORIENT=REVERSE_LANDSCAPE | Rotate page 270° |

If this option is set in the configuration file, a PCL Logical Page Orientation command is sent to the output stream before any input data. If the parameter is not recognized, no command is sent.

Line Termination Mode

The Line Termination command causes Barz_Out Pro to send a PCL Line Termination command to the printer. This instructs the PCL printer itself to perform some character substitution. For example, it is common in unix systems to have the printer add a carriage return whenever it encounters a line feed, since unix does not typically use carriage return characters in text files. For HP & PCL compatible printers only.

This Barz_Out command takes one of four parameters:

-LINE_TERM=NO_MAP	Add nothing
-LINE_TERM=CR_ADD_LF	Add LF
-LINE_TERM=LF_ADD_CR	Add CR
-LINE_TERM=BOTH	Add CR and LF

Printer Reset

This causes Barz_Out to send a PCL reset to the printer. This is usually done to restore default settings to the printer; for example, to clear a Line Termination Mode command. For HP & PCL compatible printers only.

There are actually two forms of the command:

-PCL_RESET_AT_START	Reset printer before print job
-PCL_RESET_AT_END	Reset printer after print job

USING MULTIPLE TRIGGERS

Multiple triggers allow several different barcode types or sizes to be used in one config file. Care must be taken to ensure triggers are always unique to avoid the situation where ~@ and ~@@ are both defined since the latter trigger would never be recognized. The maximum length is 30 characters (upper or lower case). The ~ and @ are the most commonly used. If the @ may appear, which is common in email addresses, you may want to use 2 characters. The combinations ~~, ~@, @~, @@ would provide 4 triggers using two characters and allowing the single characters to be barcoded (assuming the symbology being used permits those characters). A trailing trigger can be the same for all leading triggers. It is recommended to always specify the trailing trigger since the default of whitespace may appear when the document is viewed/edited, but as in the case of Postscript, the final character to be barcoded may not have any whitespace in the actual input file and thus produce undesired results.

The first trigger is defined in the cfg file, following will be a section beginning with – NEXTBARCODE command for each different trigger, i.e. barcode.

For better readability, XML style trigger sequences are highly recommended!

Example:

```
LTRIGGER=<code128>
TTRIGGER=</code128>
```

BARCODING CONTROL CHARACTERS

You may barcode control characters as permitted by the symbology being used. Code 128 is recommended. The most commonly used control characters are Tab and Enter. This allows a user to scan a field with an ending tab to move the cursor to the next field without touching the keyboard. This can be helpful for automatic input into applications that normally require manual typing and tabbing.

The format for entering hex data (control characters) is to precede the hex value with a \x.

Horizontal Tab:	\x09
Vertical Tab:	\x0B
Carriage Return:	\x0D
Line Feed:	\x0A

BARZ_OUT™ Pro TIPS

- The barcode is always printed in the spot where the first Leading Trigger character appears; the CPI setting determines the horizontal positioning.
- No more than 5 barcodes may appear on any line. Additional triggers are ignored.
- If barcodes are not printed, make sure the Leading Trigger has been placed directly in front of the data to be barcoded.
- EXAMPLE: ~483263665 is valid, however ~ 483263665 is not valid as leading trigger is two spaces before data. It should “touch”.
- Avoid placing graphic images close to the barcode data. Graphics may interfere with barcode printing.
- Do not change the Trigger character(s) to commonly used character(s), punctuation mark(s),etc.

- It is suggested to use a copy of the default switches file (sample.cfg) and modify it and save under a user or functional name such as invoice.cfg. With a CFG file you need only the CFG switch on the command line.
- To enter multiple switches, separate each with a space and include the dash character.
- If an invalid switch is entered, BARZ_OUT™ Pro's default setting will be used in its place.
- Typing *bz* at the prompt without any switch settings will use *stdio* for both input and output and should probably be avoided.

Appendix A - LISTING OF PARAMETERS

To change an option in the configuration file, add a # in front of the selection to make it a comment, remove the # to make the selection.

<u>Printer Type</u>	<u>Bin</u>	<u>Lines Per Inch</u>
-HPLASER	-AllBins	-6LPI
-EPSONFX	-UpperTray	-8LPI
-EPSONLQ	-ManualTray	
-IBMPRO9	-ManualEnv	<u>Characters Per Inch</u>
-IBMPRO24	-LowerTray	-7CPI
-POSTSCRIPT	-EnvFeeder	-10CPI
		-12CPI
<u>Print Job Options</u>	<u>Triggers</u>	-17CPI
-PAGE_ORIENT=PORTRAIT	-LTrigger=	-25CPI
-PAGE_ORIENT=LANDSCAPE	-TTrigger=	
-PAGE_ORIENT=REVERSE_PORTRAIT		<u>Text Attributes</u>
-PAGE_ORIENT=REVERSE_LANDSCAPE	<u>Human Readable</u>	-BOLD
-LINE_TERM=NO_MAP	-TextNone	-BOLD=
-LINE_TERM=CR_ADD_LF	-TextAbove	-BOLD_OFF=
-LINE_TERM=LF_ADD_CR	-TextBelow	-ITALICS
-LINE_TERM=BOTH		-UNDERLINE
-PCL_RESET_AT_START	<u>Barcode Density/Ratio</u>	-POINT_SIZE=
-PCL_RESET_AT_END	-DENSITY=	-PITCH=
	-HIGH	-REV_VIDEO_POINT
<u>Barcode Symbology</u>	-MEDIUM	-REV_VIDEO_LEN
-3_OF_9	-LOW	-REV_VIDEO_CPI
-EXT_3_OF_9	-RATIO=	
-CODABAR		<u>New Definition</u>
-CODE128	<u>Checksum</u>	-NEXTBARCODE
-2_OF_5	-NoCheckSum	
-EAN_8	-CheckSum	
-EAN_13		
-EAN-BATCH	<u>Barcode Height</u>	
-EAN_CELL	-HEIGHT=	
-EAN-EXP-DATE		
-EAN-PKG-DATE	<u>PDF417 Parameters</u>	
-EAN-PROD-DATE	-ROWS=	
-EAN-SELL-DATE	-COLUMNS=	
-EAN_SHIPISO	-DENSITY=	
-EAN_SHIPLOC	-RATIO=	
-EAN_SHIPSPA	-ASPECT=	
-PLESSEY	-TRUNCATE=	
-POSTNET	-ERRLEVEL=	
-PDF417	-IN_FILE=	
-SCC_14		
-SSCC_18		
-UCC_EAN_128		
-UPC_2		
-UPC_5		
-UPC_A		
-UPC_E		
-UPC_VERS_D		

Appendix B - PDF417 Parameters

-ROWS	Sets the number of rows, range is 3-90, default is 0 (automatic) (-ROWS=90)
-COLUMNS	Sets number of column, range 1-30, default is 0 (auto). (-COLUMNS=30).
-DENSITY	Sets cell width (x dimension), in mils, range is 10 to 100, default is 10. (-DENSITY=10 sets cell width to 10 mils).
-RATIO	Sets ratio of cell height to width, range is 1.0 to 10.0, default is 3.0 ("-RATIO=3.2" sets the cell height to 3.2 times cell width, for a ratio of 3.2:1).
-ASPECT	Sets the "width" of the aspect ratio, which is " <i>height:width</i> " of the entire barcode. The height is fixed at '1'. The range is 0.1 to 99.9, the default is 2.0. "-ASPECT=2.0" would result in an aspect ratio of '1:2'.
-TRUNCATE	Truncate the barcode. -TRUNCATE=Y selects a truncated barcode; 'N' selects a normal, non-truncated bar. The default is normal.
-ERRLEVEL	Selects error level. Range is 0 to 8, default is 0.
-IN_FILE	Determines whether to interpret the data string parameter (-datafilename) as a file name or as a string to be encoded. IN_FILE=Y means take the string as a file name, and pass the contents of the file to the encoder. IN_FILE=N means interpret the data string as a string and pass that string to the encoder. The default is 'N', take the data as a string.

These parameters are what we use to define a PDF417 barcode across all Unibar products. They are a subset of the parameters defined by the PDF417 symbology.

- **X dimension** – width of a module, or cell, specified in mils, range is 1 to 100, default is 10.
- **Y dimension** – height of a module, specified as a multiple of the x dimension (sometimes referred to as a ratio; e.g., "3.0:1"). The range is 1.0 to 10.0; the default is 3.0.
- **Number of Rows**– number of rows of modules, range is 3 to 90. A value of 0 means 'automatic', let the printer determine the number of rows needed. The default is automatic.
- **Number of columns**– number of columns of codewords, range is 1 to 30. A value of 0 means automatic, like rows. The default is automatic.
- **Error Level** – The range is 0 to 8, default is 0.
- **Truncate** – Yes or no, default is no.
- **Aspect** – Aspect ratio of the barcode, height to width. The height is set as '1', the range of the width is 0.1 to 100. The default is 1:2.

Handling of Rows, Columns and Aspect Ratio of the overall Barcode

By default, the barcode is made as small as possible for the given data, with an overall aspect ratio of 1:2. That is, height to width of the entire barcode.

If the user sets the Row value, the number of rows is fixed at that value, the number of columns is set to *automatic* and the barcode will add columns as needed to handle the given data.

Likewise, if the user sets the Column value, the number of rows is set to automatic.

Thus, the user has a way to control either the width or the height of the barcode. The other direction grows depending on the amount of data.

If the user sets both the Row and Column, the size is fixed. ***The user must make sure the data will fit in the specified barcode size.***

The aspect ratio is only used if both rows and columns are left at, or set to, zero.

This table shows the support provided.

Print Device	x-dim	y-dim	Rows	Columns	Error Level	Truncate	Aspect
PCL Laser	3-100, 3.33	2.0-4.0, 0.1	Y	Y	Y	Y	

Appendix C - EAN Application Identifiers

Data Content	AI	Plus the data	Config File	Human Readable
Serial Shipping Container Code (SSCC-18)	00	exactly 18 digits	-SSCC_18	(00) 0 0614141 000000011 1
Shipping Container Code (SCC-14)	01	exactly 14 digits	-SCC_14	(01) 1 0 614141 00041 5
Batch Numbers	10	up to 20 alphanumeric	-EAN-BATCH	(10) 12345678901234567890
Production Date (YYMMDD)	11	exactly 6 digits	-EAN-PROD-DATE	(11) 001230
Package Date (YYMMDD)	13	exactly 6 digits	-EAN-PKG-DATE	(13) 001230
Sell By Date (YYMMDD)	15	exactly 6 digits	-EAN-SELL-DATE	(15) 001230
Expiration Date (YYMMDD)	17	exactly 6 digits	-EAN-EXP-DATE	(17) 001230
Ship To Location Code	410	exactly 13 digits	-EAN_SHIPLOC	
Ship To Postal Code within Single Postal Authority	420	up to 9 alphanumeric	-EAN_SHIPSPA	(420) 483093103
Ship To Postal Code with 3-digit ISO Country Code Prefix	421	3 digits plus up to 9 alphanumeric	-EAN_SHIPISO	(421) ISOabcdefghi
Electronic Serial number for cellular phone	8002	up to 20 alphanumeric	-EAN_CELL	(8002)

In addition, these same Application Identifiers, and many more, along with their appropriate data can be “chained” together in GS1-128 (formerly UCC/EAN-128) barcodes.

Code	Description	Content
00	Serial Shipping Container Code(SSCC-18)	18 digits - numeric
01	Shipping Container Code (SSCC-14)	14 digits - numeric
02	Number of containers	14 digits - numeric
10	Batch Number	1-20 alphanumeric
11	Production Date	6 digits: YYMMDD
13	Packaging Date	6 digits: YYMMDD
15	Sell by Date (Quality Control)	6 digits: YYMMDD
17	Expiration Date	6 digits: YYMMDD
20	Product Variant	2 digits
21	Serial Number	1-20 alphanumeric
22	HIBCC Quantity, Date, Batch and Link	1-29 alphanumeric
23x	Lot Number	1-19 alphanumeric
240	Additional Product Identification	1-30 alphanumeric
250	Second Serial Number	1-30 alphanumeric
30	Quantity Each	
310y	Product Net Weight (kg)	6 digits
311y	Product Length/1st Dimen (meters)	6 digits
312y	Product Width/Diameter/2nd Dim. (meters)	6 digits
313y	Product Depth/Thickness/3rd Dim> (meters)	6 digits
314y	Product Area, (square meters)	6 digits
315y	Product Volume, (liters)	6 digits
316y	product Volume, (cubic meters)	6 digits
320y	Product Net Weight, (pounds)	6 digits
321y	Product Length/1st Dimension, (inches)	6 digits
322y	Product Length/1st Dimension, (feet)	6 digits
323y	Product Length/1st Dimension (yards)	6 digits
324y	Product Width/Diameter/2nd Dim. (inches)	6 digits
325y	Product Width/Diameter/2nd Dim. (feet)	6 digits
326y	Product Width/Diameter/2nd Dim. (yards)	6 digits
327y	Product Depth/Thickness/3rd Dim. (inches)	6 digits
328y	Product Depth/Thickness/3rd Dim. (feet)	6 digits
329y	Product Depth/Thickness/3rd Dim. (yards)	6 digits

Code	Description	Content
330y	Container Gross Weight (Kg)	6 digits
331y	Container Length/1st Dimension (Meters)	digits
332y	Container Width/Diameter/2nd Dim. (Meters)	6 digits
333y	Container Depth/Thickness/3rd Dim. (Meters)	6 digits
334y	Container Area (Square Meters)	6 dig its
335y	Container Gross Volume (Liters)	6 digits
336y	Container Gross Volume (Cubic Meters)	6 digits
340y	Container Gross Weight (Pounds)	6 digits
341y	Container Length/1st Dimension (inches)	6 digits
342y	Container Length/1st Dimension (feet)	6 digits
343y	Container Length/1st Dimension (yards)	6 digits
344y	Container Width/Diamater/2nd Dim. (inches)	6 digits
345y	Container Width/Diameter/2nd Dim. (feet)	6 digits
346y	Container Width/Diameter/2nd Dim. (yards)	6 digits
347y	Container Depth/Thick/Ht/3rd Dim.(inches)	6 digits
348y	Container Depth/Thick/Ht/3rd Dim. (feet)	6 digits
349y	Container Depth/Thick/Ht/3rd Dim. (yards)	6 digits
350y	Product Area (Square Inches)	6 digits
351y	Product Area (Square Feet)	6 digits
352y	Product Area (Square Yards)	6 digits
353y	Container Area (Square Inches)	6 digits
354y	Container Area (Square Feet)	6 digits
355y	Container Area (Suqare Yards)	6 digits
356y	Net Weight (Troy Ounces)	6 digits
360y	Product Volume (Quarts)	6 digits
361y	Product Volume (Gallons)	6 digits
362y	Container Gross Volume (Quarts)	6 digits
363y	Container Gross Volume (Gallons)	6 digits
364y	Product Volume (Cubic Inches)	6 digits
365y	Product Volume (Cubic Feet)	6 digits
366y	Product Volume (Cubic Yards)	6 digits
367y	Container Gross Volume (Cubic Inches)	6 digits
368y	Container Gross Volume (Cubic Feet)	6 digits
369y	Container Gross Volume (Cubic Yards)	6 digits
37	Number of Units Contained	1-8 digits
400	Customer Purchase Order Number	1-29 alphanumeric
410	Ship To/Deliver To Location Code	13 digits
411	Bill To/Invoice Location Code (EAN13 or DUNS code)	13 digits
412	Purchase From Location Code (EAN13 or DUNS code)	13 digits
420	Ship To/Deliver To Postal Code (Single Postal Authority)	1-9 alphanumeric
421	Ship To/Deliver To Postal Code (Multiple Postal Authority)	4-12 alphanumeric
8001	Roll Products - Width/Length/Core Diameter	14 digits
8002	Electronic Serial Number (ESN) for Cellular Phone	1-20 alphanumeric
8003	UPC/EAN Number and Serial Number of Returnable Asset	14 Digit UPC +1-16 Alphanumeric Serial Number
8004	UPC/EAN Serial Identification	1-30 Alphanumeric
8005	Price per Unit of Measure	6 digits
8100	Coupon Extended Code: Number System and Offer	6 digits - numeric
8101	8101 Coupon Extended Code: Number System, Offer,	10 digits - numeric

Code	Description	Content
	End of Offer	
8102	Coupon Extended Code: Number System preceded by 0	2 digits - numeric
90	Mutually Agreed Between Trading Partners	1-30 alphanumeric
91	USPS services	2-digit service code, 9-digit customer ID, 8-digit package ID plus 1 Mod10 check digit
92	Internal Company Codes	1-30 alphanumeric
93	Internal Company Codes	1-30 alphanumeric
94	Internal Company Codes	1-30 alphanumeric
95	Internal Company Codes	1-30 alphanumeric
96	Internal Company Codes	1-30 alphanumeric
97	Internal Company Codes	1-30 alphanumeric
98	Internal Company Codes	1-30 alphanumeric
99	Internal Company Codes	1-30 alphanumeric

Notes:

- Use -UCC_EAN_128 in the configuration file to select GS1-128 barcode type.
- Data sequence must start with an AI code.
- For multiple AI codes in a sequence, use !1 (for FNC1) to indicate that AI follows.
Example: 9112345!14209876543!13693100
Would produce
Barcode content: <FNC1>9112345<FNC1>4209876543<FNC1>3693100
Human Readable: (91)12345(420)9876543(3693)100
FNC1 is Function Control Character 1
- AI code table values with x or y indicate number of digits to follow.
Example: 320512345

Appendix D – Examples of Configuration Files

Several sample configuration files are included with BARZ_OUT. Each one has a corresponding text or input file to go with it.

sample.cfg – all options defined, but commented out. You can modify copies of this file.

symbolTriggers.cfg – uses combinations of symbols such as ~ and @

textAttributes.cfg – definitions to show effects such as bold and italics

xmlTriggers – definitions with XML style instead of symbols

samplePS.cfg – XML style triggers for use with the included samplePS.ps postscript input file

Appendix E – Installing BARZ_OUT™ as a CUPS filter

In order to run BARZ_OUT as a CUPS filter in a Unix/Linux environment, certain files must be copied to the proper locations as follows:

- o bz (executable) in /usr/lib/cups/filter
- o unibar.key (license key file) in /usr/lib/cups/filter *requires Enterprise License with STDOUT*
- o bz.conf – (configuration file) in /etc/cups

The mime.types and mime.convs configuration files used by the CUPS print server should not be modified. These files may be overwritten by the system. For our example filter, we will use a suffix of .ubt for text files on which we wish to add barcodes. The 'bz.types' file included with BARZ_OUT must be copied to /etc/cups. Its contents are:

```
# Local MIME definition for Unibar's BARZ_OUT text files
application/x-barz_out-text ubt printable(0,1024)
```

The 'bz.convs' file included with BARZ_OUT must also be copied to /etc/cups. Its contents are:

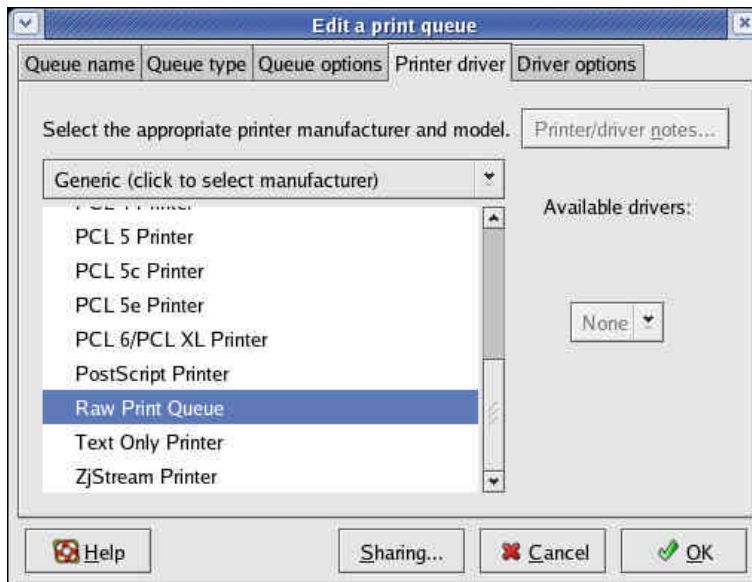
```
# Local filter for Unibar's BARZ_OUT text files
application/x-barz_out-text application/vnd.cups-raw 1 bz
```

This basically says that text files with a .ubt extension are converted to raw printer commands by way of the bz filter program. Once these files are in place, the CUPS system must be “bounced” as follows:

```
/etc/init.d/cups restart
```

After completing the steps above, any text file with a .ubt extension submitted for printing via the 'lp' command will automatically be processed by the BARZ_OUT application.

It is important to note that CUPS filters are not applied on raw queues. Since the output generated by Unibar applications such as BARZ_OUT and BARCODE 2000 is already in the printer's native language, this output is most often sent to a raw queue for normal operation.



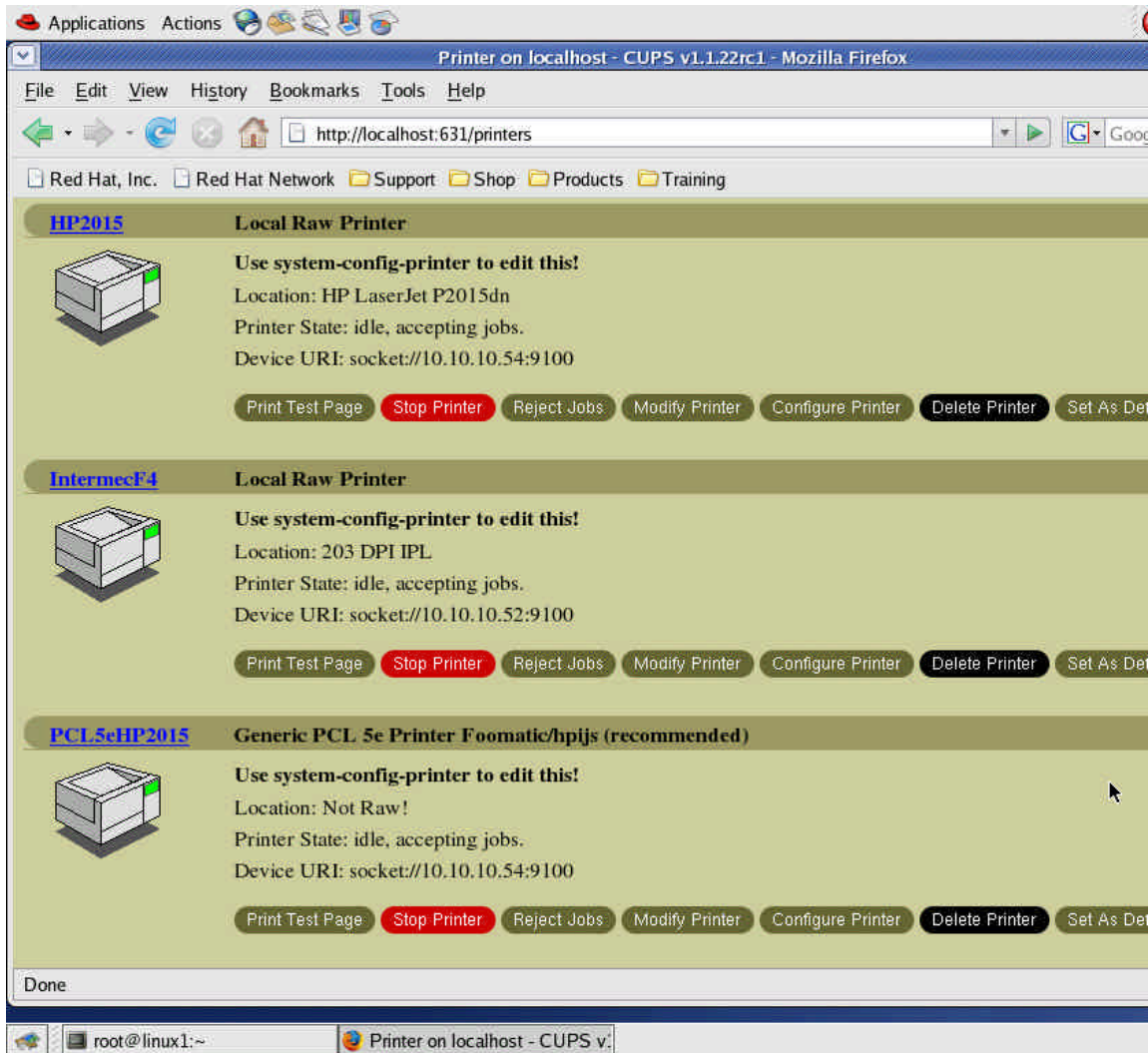
However, since CUPS will not call a filter program if the destination queue is raw, we must send any .ubt files to a “regular” print queue which has a driver.



In this example, our printer really is a PCL5e printer, but the hpjjs driver will not actually be used for any .ubt file since it is converted to raw by the BARZ_OUT filter. It is suggested that you first test BARZ_OUT on the command line to a raw queue. Once it works properly, copy the .cfg BARZ_OUT configuration file used in successful testing to /etc/cups as bz.conf (to follow the CUPS naming conventions). Then, simply send the .ubt print jobs to a “non-raw” queue.

```
Test BARZ_OUT:      bz -CFGxmlTriggers.cfg < xmlTriggers.txt | lp -d HP2015
Test bz CUPS Filter: lp -d PCL5eHP2015 sample.ubt
```

Please note that xmlTriggers.txt and sample.ubt have the same contents. Also, the bz.conf used with our CUPS has the same contents as xmlTriggers.cfg. If you can run a web browser on the system where the print queues were created, you can use the CUPS Web Interface to check them by pointing your browser at <http://localhost:631/printers>. To use the Web Interface you will need the login/passwd of a user that is allowed to configure printers, which may be root. Please see screenshot of the two queues in our example on the next page.



(You may disregard the Intermecc F4 queue created for use with Unibar's BARCODE 2000 Label printing application.)

The following is an excerpt from `/var/log/cups/error_log` pertaining to our test above:

```
D [09/Jan/2008:08:56:38 -0500] AcceptClient: 5 from localhost:631.
D [09/Jan/2008:08:56:38 -0500] ReadClient: 5 POST /printers/PCL5eHP2015 HTTP/1.1
D [09/Jan/2008:08:56:38 -0500] print_job: auto-typing file...
D [09/Jan/2008:08:56:38 -0500] print_job: request file type is application/x-barz_out-text.
D [09/Jan/2008:08:56:38 -0500] CancelJob: id = 408
D [09/Jan/2008:08:56:38 -0500] check_quotas: requesting-user-name = 'applmgr'
D [09/Jan/2008:08:56:38 -0500] print_job: requesting-user-name = 'applmgr'
I [09/Jan/2008:08:56:38 -0500] Adding start banner page "none" to job 908.
I [09/Jan/2008:08:56:38 -0500] Adding end banner page "none" to job 908.
I [09/Jan/2008:08:56:38 -0500] Job 908 queued on 'PCL5eHP2015' by 'applmgr'.
D [09/Jan/2008:08:56:38 -0500] Job 908 hold_until = 0
D [09/Jan/2008:08:56:38 -0500] StartJob(908, 0x9f274e0)
D [09/Jan/2008:08:56:38 -0500] StartJob() id = 908, file = 0/1
D [09/Jan/2008:08:56:38 -0500] job-sheets=none,none
D [09/Jan/2008:08:56:38 -0500] banner_page = 0
D [09/Jan/2008:08:56:38 -0500] StartJob: argv =
"PCL5eHP2015", "908", "applmgr", "sample.ubt", "1", "", "", "/var/spool/cups/d00908-001"
D [09/Jan/2008:08:56:38 -0500] StartJob:
envp[0]="PATH=/usr/lib/cups/filter:/bin:/usr/bin"
```

```

D [09/Jan/2008:08:56:38 -0500] StartJob: envp[1]="SOFTWARE=CUPS/1.1"
D [09/Jan/2008:08:56:38 -0500] StartJob: envp[2]="USER=root"
D [09/Jan/2008:08:56:38 -0500] StartJob: envp[3]="CHARSET=utf-8"
D [09/Jan/2008:08:56:38 -0500] StartJob: envp[4]="LANG=en_US"
D [09/Jan/2008:08:56:38 -0500] StartJob: envp[5]="PPD=/etc/cups/ppd/PCL5eHP2015.ppd"
D [09/Jan/2008:08:56:38 -0500] StartJob: envp[6]="CUPS_SERVERROOT=/etc/cups"
D [09/Jan/2008:08:56:38 -0500] StartJob: envp[7]="RIP_MAX_CACHE=8m"
D [09/Jan/2008:08:56:38 -0500] StartJob: envp[8]="TMPDIR=/var/spool/cups/tmp"
D [09/Jan/2008:08:56:38 -0500] StartJob: envp[9]="CONTENT_TYPE=application/x-barz_out-text"
D [09/Jan/2008:08:56:38 -0500] StartJob: envp[10]="DEVICE_URI=socket://10.10.10.54:9100"
D [09/Jan/2008:08:56:38 -0500] StartJob: envp[11]="PRINTER=PCL5eHP2015"
D [09/Jan/2008:08:56:38 -0500] StartJob: envp[12]="CUPS_DATADIR=/usr/share/cups"
D [09/Jan/2008:08:56:38 -0500] StartJob: envp[13]="CUPS_FONTPATH=/usr/share/cups/fonts"
D [09/Jan/2008:08:56:38 -0500] StartJob: envp[14]="CUPS_SERVER=localhost"
D [09/Jan/2008:08:56:38 -0500] StartJob: envp[15]="IPP_PORT=631"
D [09/Jan/2008:08:56:38 -0500] StartJob: statusfds = [ 9 10 ]
D [09/Jan/2008:08:56:38 -0500] StartJob: filterfds[1] = [ 11 -1 ]
D [09/Jan/2008:08:56:38 -0500] StartJob: filter = "/usr/lib/cups/filter/bz"
D [09/Jan/2008:08:56:38 -0500] StartJob: filterfds[0] = [ 12 13 ]
D [09/Jan/2008:08:56:38 -0500] start_process("/usr/lib/cups/filter/bz", 0xbfeeaa40, 0xbfee9db0, 11, 13, 10)
I [09/Jan/2008:08:56:38 -0500] Started filter /usr/lib/cups/filter/bz (PID 17286) for job 908.
D [09/Jan/2008:08:56:38 -0500] StartJob: backend = "/usr/lib/cups/backend/socket"
D [09/Jan/2008:08:56:38 -0500] StartJob: filterfds[1] = [ -1 11 ]
D [09/Jan/2008:08:56:38 -0500] start_process("/usr/lib/cups/backend/socket", 0xbfeeaa40, 0xbfee9db0, 12, 11, 10)
I [09/Jan/2008:08:56:38 -0500] Started backend /usr/lib/cups/backend/socket (PID 17287) for job 908.
D [09/Jan/2008:08:56:38 -0500] ProcessIPPRequest: 5 status_code=0
D [09/Jan/2008:08:56:38 -0500] ReadClient: 8 POST /printers/PCL5eHP2015 HTTP/1.1
D [09/Jan/2008:08:56:38 -0500] ProcessIPPRequest: 8 status_code=0
D [09/Jan/2008:08:56:38 -0500] CloseClient: 5
D [09/Jan/2008:08:56:38 -0500] ReadClient: 8 POST /printers/PCL5eHP2015 HTTP/1.1
D [09/Jan/2008:08:56:38 -0500] ProcessIPPRequest: 8 status_code=0
D [09/Jan/2008:08:56:38 -0500] [Job 908] BZ: barz_out application running.
D [09/Jan/2008:08:56:38 -0500] [Job 908] BZ: argv[0] = PCL5eHP2015
D [09/Jan/2008:08:56:38 -0500] [Job 908] BZ: argv[1] = 908
D [09/Jan/2008:08:56:38 -0500] [Job 908] BZ: argv[2] = applmgr
D [09/Jan/2008:08:56:38 -0500] [Job 908] BZ: argv[3] = sample.ubt
D [09/Jan/2008:08:56:38 -0500] [Job 908] BZ: argv[4] = 1
D [09/Jan/2008:08:56:38 -0500] [Job 908] BZ: argv[5] =
D [09/Jan/2008:08:56:38 -0500] [Job 908] BZ: argv[6] = /var/spool/cups/d00908-001
D [09/Jan/2008:08:56:39 -0500] UpdateJob: job 908, file 0 is complete.
D [09/Jan/2008:08:56:39 -0500] CancelJob: id = 908
D [09/Jan/2008:08:56:39 -0500] StopJob: id = 908, force = 0
D [09/Jan/2008:08:56:39 -0500] StopJob: printer state is 3

```