

Trans-Micro Express32™

Trans-Micro, Inc.
PO Box 898
Fairfield, FL 32634
(800) 749-2335 Sales
(352) 591-1030
(352) 591-1015 Tech Support
(352) 591-1096 FAX
www.trans-micro.com

Version 3.00 Beta Documentation

01/04/1999

Copyright 1999, Trans-Micro, Inc.

All Rights Reserved
No portion of this document may be reproduced by any means unless
authorized in writing by Trans-Micro, Inc.

Contents

| | |
|---|----|
| License | 1 |
| Welcome to Express32 | 2 |
| A note about our manual and our software. | 2 |
| What is Express32? | 3 |
| How do I make it all work? | 4 |
| Installation | 5 |
| Making it Work | 6 |
| Creating Graphic Images | 6 |
| Creating Form Overlays | 6 |
| Adding Express32 commands to your application | 7 |
| Using Express32 Commands in Windows Applications | 7 |
| A Note About Laser Printer Memory | 9 |
| Express32, Two Modes of Operation | 10 |
| Polling Mode: | 10 |
| Command Line Mode: | 10 |
| Express32 Command Line Options. | 10 |
| Express32 Console Menu (Polling Mode) | 11 |
| 1> Test destination printer. | 11 |
| 2> Change destination printer. | 11 |
| 3> Process a print file. | 11 |
| 4> Reprocess last print job. | 11 |
| 5> Print Express Error Log. | 11 |
| 6> About Express32 | 11 |
| 7> Micr Alignment | 12 |
| Sample Form Overlays | 14 |
| The EXPRESS.LOG File | 15 |
| The EXPRESS.INI file | 16 |
| The EX32USER.INI file | 18 |
| Express32 Command Files | 19 |
| Standard CMD files: | 19 |
| Special CMD files: | 19 |

| | |
|--|----|
| Print Job Rules | 20 |
| Command Structure | 21 |
| Multiple command example: | 22 |
| Command Measurement Units | 23 |
| UNITS: | 23 |
| Command Reference | 24 |
| Search Order for Files used with Express32 | 24 |
| Files with PCX or PCL file extensions:. | 24 |
| Files with FRM or PRN file extensions: | 24 |
| All other file extensions, no path included: | 24 |
| CMDFILE | 25 |
| OVERLAY | 26 |
| GRAPHIC | 27 |
| PAGE32 | 29 |
| PAGE (obsolete) | 30 |
| BINS | 31 |
| COPIES | 32 |
| PRINT@ | 33 |
| TEXTBOX | 35 |
| TEXTBOXFONT | 38 |
| PCL | 39 |
| CODE39. | 40 |
| STARTCHECK | 42 |
| MICRPAGE. | 43 |
| SIGLINE. | 46 |
| MICRAMOUNT | 47 |
| MICRLINE | 48 |
| ENDCHECK | 50 |
| MICR Symbol Translations | 51 |
| Command Order | 52 |
| Debugging | 53 |
| Final Notes: | 53 |
| APPENDIX A | 54 |
| A word about printing MICR documents: | 54 |
| What is a MICR check? | 54 |
| Why is the MICR line so important? | 54 |

Express32

Transit Number: 54
On Us Number: 55
Additional Information: 55

License

Definition of Software as applies to this license:

Software is defined as the executable programs distributed in this package.

This software is licensed for use on one computer only. You may make copies of this software for achival purposes only. You may not copy this software and give, lend, or distibute it to anyone other than the original purchasor. Trans-Micro will prosecute anyone found to possess illegal copies of its software.

This software may not be altered or modified in any fashion except as expressly allowed by Trans-Micro, Inc..

If you plan on distributing the EXPRESS32 driver with software applications that you have developed, please contact Trans-Micro for information regarding licensing the driver for distibution with your application.

Express32

Welcome to Express32

Trans-Micro's Express32 tools you have the absolute easiest way to print Form Overlays, Graphics and MICR checks on your HP laser printer. Express32 is a print file processor

A note about our manual and our software.

It may seem that our manual is a bit on the small side. This is done intentionally. We feel that the size of instruction manual is inversely proportional to the quality of the software. In other words if the software is designed properly then it should be fairly obvious how to use it. The software should be free from those little "GOTCHA's" that a lot of manuals warn you about. The commands and their structure are the most important things you need to know about Check Factory Express32. Our software performs just a few functions but we think it does them quite well.

What is Express32?

The Express32 program is an extremely fast and compact 32 bit console application. It must run in a Windows 9x or NT window. It will not run under 16 bit DOS or Windows 31. Express32 will print to any available windows printer anywhere on your network.

When you send a print job to Express32 the program looks for Express32 commands embedded in your print job. When it finds a command, it strips it from the print job, and then performs that function. An example would be uploading a graphic to the printer or printing the MICR line on a check document. You may also place some commands in the EXPRESS.INI file or in separate command files.

You can instantly add Form Overlays and Graphics to your application with absolutely NO PCL programming required. Express32 takes care of all the messy details when dealing with creating and uploading form overlays and graphics. Forms and graphics are only uploaded to the printer one time, saving time and drastically reducing printing times. You can give your applications that professional look with "typeset quality" forms on plain blank paper.

Express32 is the ideal solution for MICR check printing. You may print multiple checks per page and not have to worry about calculating the logical print positions for everything on each check. Express32 takes care of all the calculations instantly. The MICR line is always formatted correctly and printed in the right place.

How do I make it all work?

As a developer all you have to do is:

1> Design your application's reports to print your documents to the laser printer. Make sure that report pages correctly on the laser printer.

2> Next you create a form overlay in your favorite publishing program or create an overlay as an Express32 command file using our TextBox and Print@ commands. See "Creating Form Overlays". Make sure the form overlay lines up with your application print out. Create any PCX graphic images that may be needed.

3> Add the necessary Express32 commands to each page of the report. They are simple delimited ASCII commands that are read by the Express32 program. The commands are stripped from your print job and replaced by the function that you have told Express32 to perform. It couldn't be easier!

4> Let Express32 process the print job. It will seamlessly merge your print output with graphics with out any complicated graphics programming.

Installation

Insert the floppy in to your A: drive. Run A:SETUP.EXE. Select your drive and path to install to. We recommend that you install Express32 to a EXPRESS folder below your top level application folder.

Example: C:\MYAPP\EXPRESS

Make sure you have a HP PCL5 compatible laser printer (HP III or above) available and connected to a windows printer. If your printer is not PCL5 compatible some PRINT@ functions may not work correctly.

Making it Work

Creating Graphic Images

Express32 can only read black and white (2 color) PCX graphics files. It cannot deal with color or greyscale type files.

When creating graphic images using a paint or scanner program please save the image as 300 dpi black and white. Some programs refer to black and white as "Line Drawing".

Save the PCX graphic to the EXPRESS\GRAPHICS directory.

Never export a color image. Color images cannot be converted by Express32.

Creating Form Overlays

Form overlays can be created in any publishing or word processing program that can send its' output to a Windows HP PCL printer driver.

1> Install Windows HP PCL5 printer.

Install a HP Laserjet III printer driver on your windows system. Point the driver to your laser printer. We are assuming that you are working with a PCL5 type laser printer and not PCL4 (HP SeriesII).

2> Create Form Overlay

Create your overlay as a one page document. Print the page to your laser printer using the above printer. If the output is satisfactory go on to step 3.

3> Print Form Overlay to File.

Next you will print the above overlay to a file using the HP Laserjet III driver. If the program that you are using has a "Print to File" option box in the Print Window just check the box and print the one page document to a file name in the EXPRESS\FORMS directory. Use a PRN or FRM extension to indicate that it is a form overlay file. Example: MYFORM.FRM.

You may now use this form overlay in the Express32 OVERLAY or GRAPHICS command.

Adding Express32 commands to your application

The last thing to do is to add the Express32 commands to your application printout. This is done by placing the commands in your report design screen or by adding programming code. This step is specific to your application so we are not able to tell you exactly how to perform this step.

Using Express32 Commands in Windows Applications

If you are adding Express32 commands in a windows application you must use PRINTER RESIDENT FONT for the Express32 commands. In many windows font selection popups, a printer resident font is noted by a printer symbol to the left of the font name as opposed to a TrueType symbol. When Express32 processes the print job it cannot read Express32 commands if they are in a TrueType font.

In order to select a laser printer resident font you must have a HP Laserjet III or IV driver set as the default printer on your system when you select the font. We strongly recommend that you select the LINEPRINTER font when printing EXPRESS32 commands. All other portions of the report may be in any font available.

Test Print Your Report

Next run your report WITHOUT processing it through Express32. Make sure that the report still pages in the correct places. Also check the Express32 commands for any errors, as they will print on the page.

Run Express32 in Polling Mode

Run the Express32 program in polling mode by clicking on Start, Express32, EXP32

From the Express32 console enter the correct Destination printer and select Test Destination printer from the menu.

Express32

Run your application and print the report to C:\EXPRESS.PRN. If you have used the correct syntax for the Express32 commands your report will print with the form graphics and data merged on each page. If you hear a beep while printing please check the EXPRESS.LOG file in the local workstation's WINDOWS directory. This will contain any error messages generated by Express32 while printing the job. The messages should be self explanatory. The log file may be cleared each time Express32 is started.

A Note About Laser Printer Memory

Depending on the HP laser driver you select, your overlay file may be quite large and in fact exceed the amount of memory in your laser printer. Express32 stores forms and graphics as macros in the laser printer. If you are creating overlays for distribution to customers you must take into consideration what type of laser printer they will be using. All overlays must then be printed to file using the lowest common laser printer driver. Note that older laser printer drivers create much larger print files than ones for the HP III and IV series.

The drivers for the HP III and IV series compress the data and will produce smaller print files BUT the files will ONLY print on their respective printers! An HP IV print file will not print on a HP III printer.

Express32

Express32, Two Modes of Operation

Polling Mode:

When running in Polling Mode, Express32 looks at the file defined as the SourceFile, usually located at C:\EXPRESS.PRN

Express32 checks this file about one a second to see if there is data in it. If there is Express32 processes the file and sends it to the Destination Printer.

The Source File name may be changed in EXPRESS.INI.

Command Line Mode:

Express32 can be used from within your application by executing the program with a command line argument(s).

In Visual Basic for example you might use the VB SHELL command to have Express32 process the file MYINVOICE.PRN

```
rc = SHELL("C:\MYAPP\EXPRESS\EXP32.EXE _  
/FILE=C:\MYAPP\MYINVOICE.PRN", 5)
```

Other languages have similar methods of running outside programs.

You may also wish to create a separate EXPRESS.INI file for each file or type of file that you process. If you create a copy of the EXPRESS.INI file in the EXPRESS folder, with the same base file name as the file being processed, Express32 will use that INI file instead of EXPRESS.INI. This opens up a world of possibilities.

You may also specify a different EXPRESS.INI on the command line using /INIFILE=MYINI.INI

Express32 Command Line Options

/File = Path\Filename.EXT

/DestPrinter = LPT1, LPT2, LPT3, \\AnyServer\AnyHPPrinter

/IniFile = AnyExpressIniFile.INI

/Help

Express32 Console Menu (Polling Mode)

Run Express32 from the Start Menu, Express32, EXP32. After the sign on Express32 will go into Polling Mode and you will see the following menu:

1> Test destination printer.

Sends a short text message to the currently selected destination printer and then ejects the page. Used to verify that the printer is online and connected.

2> Change destination printer.

Allows you to change the destination printer to any available windows printer. Please send Express32 output to HP COMPATIBLE LASER PRINTERS ONLY!

3> Process a print file.

Used for processing a printfile manually. Usefull for testing print files containing Express32 commands generated somewhere else and sent to this workstation or for processing test files.

4> Reprocess last print job.

Each print file is stored in C:\LASTFILE.000 after being processed. Should you choose to reprocess the file through Express32 again due to errors etc. It is available here without having to regenerate it through your application.

5> Print Express Error Log.

Prints the Express32 error log to the currently selected destination printer.

6> About Express32

Copyright infomation about Express32 and information about Trans-Micro, Inc.

Express32

7> Micr Alignment

Aligns the current printer for Micr check printing. Saves the alignment information in the EX32USER.INI file located in the workstations' WINDOWS folder.

Load the check paper tray with 8 1/2" x 11" paper. You do not need to use check paper to do the alignment as long as the plain paper and check paper are the same size.

Select ALIGN MICR from the console menu. Follow the prompts.

The MICR line will ALWAYS print at the bottom of the page no matter how you may intend to print the checks through your application. This alignment is to determine a relative position for this paper bin.

Assuming you have provided the Express32 MicrPage command with the correct Spacing, Checks per Page, and Bottom Margin parameters, your MICR line will always be positioned perfectly. Please note that the MICR line may float around a little because of paper movement in the tray. Also, printing checks using the manual feed tray on some laser printers may cause serious misalignment horizontally. Please print test runs using the intended paper tray and check your checks with the alignment guage.

Check Factory Express32 is preset to be used with an HP Laserjet III laser printer. Other laser printers have slightly different printing locations. The positioning of the MICR line is critical. Run the ALIGN program to print a sample MICR line. Use the alignment gauge provided to determine if your MICR line is in the correct position.

Express32

When using our BINS command to pull paper from a paper source bin other than the default bin, you must align your checks in the bin from which the check forms will be pulled. The ALIGN program will prompt you for a bin number. For a Hewlett Packard IV printer BIN 1 is the 200 sheet cassette, BIN 4 is the upper paper tray with the fold out door and BIN 5 is the lower 500 sheet tray. BIN 2 is almost always the manual feed tray. Please see your printer manual for bin numbers of other printers.

If the line needs adjustment enter changes to the Vertical and Horizontal offsets until the MICR line falls within the tolerances of the gauge. Remember that the alignment adjustments are in DECIPOINTS! There are 720 decipoints to one inch.

Sample Form Overlays

There are a number of sample form overlays provided for use in your own applications. Most are check formats of various types. The files end in .FRM and the names are descriptive. For example BCHK1UPM.FRM is a one up business check in the middle of the page.

Use Express32 in command line mode to print them out. Example: From the Express32 console menu select "Process a Print File". Enter BCHK1UPM.FRM when prompted for a file name.

The EXPRESS.LOG File

By design, Express32 does not display messages during its operation. All errors encountered during operation are printed to the EXPRESS.LOG file in the local workstations' WINDOWS directory. You may wish to check the log file after each print operation to see if there are any errors.

If you hear a beep while printing or your output contains unexpected results, please check the EXPRESS.LOG file. This will contain any error messages generated by Express32 while printing the job. The messages should be self explanatory.

The log file may be cleared each time Express32 is started by setting `ClearLogOnStart=1` in the [User] section of the EX32USER.INI file in the workstations' WINDOWS folder.

Express32

The EXPRESS.INI file

EXPRESS.INI is the default INI file. Express32 is designed to use multiple INI files. If you print using Express32 in command line mode Express32 will look for an INI file in the EXPRESS folder with the same Base file name as the file name being processed. If it cannot find that INI file it will use the EXPRESS.INI instead.

Example:

Command line: EXP32 /File=MyFile.PRN

Express32 will try to use MyFile.INI if it is available.

Using Express32 in this manner could allow you to print jobs with overlays and graphics without any embedded Express32 commands. The [FirstPage] and [EachPage] sections could be used to enter all the Express32 commands that would be needed for this print job, using little or no embedded commands.

All text after the semi colon character is regarded as remarks and comment and is not read by Express32.

[Program] Section:

MaxLogSize = 10000

If the EXPRESS.LOG file size exceed this value, it will cease to record more log events.

;MaxLineFeedsPerPage = 60

Uncomment this line if you find it impossible to cause a form feed between each page of a file that Express32 is to process. Set this value to the exact number of CRLF's per page. This is not usefull for processing print files generated by windows.

[MicrPage] Section:

;Sets the default values for printing MICR checks

ChecksPerPage = 1

CheckHeight = 11 ;(Spacing)

CheckBottomMargin = 0

MicrFormatDefault = B

CheckNoFormatDefault = ##### ; or 00000000 #'s and 0's may be used interchangeably.

The next two sections allow you to perform Express32 commands on a print job without having to embed commands within the print job or alter it in any way.

[FirstPage] Section:

Commands to execute only on the first page of the print job.

Any Express32 command may be used.

; Example

Graphic, MyLogo.PCX, 0.3, 0.3

[EachPage] Section:

Commands to execute on EACH page of the print job.

Any Express32 command may be used.

; Example

Graphic, MySig.PCX, 5, 6.7

[AutoMicr] Section:

The auto MICR section allows you to print numerically sequenced MICR checks without having to add any embedded commands in to a print file.

TransitDefault = {the check Transit Number - Bank Routing Number}

OnUsDefault = {the check OnUs number - Account Number}

CheckNumberDefault = { the starting check number} (The check number will be incremented by one for each check printed.

Express32

The EX32USER.INI file

A EX32USER.INI is located in the WINDOWS folder of each workstation that uses Express32. This file stores the workstation specific settings that Express32 needs to run on each workstation.

[User] Section:

DestPrinter = {currently selected destination printer}

SourceFile = C:\EXPRESS.PRN

File that Express32 polls for when in polling mode.

DestFile = C:\DESTFILE.000

File that Express32 copies SourceFile to when it finds Source File. Express32 then actually processes DestFile.

LastFile = C:\LASTFILE.000

After Express32 finishes processing DestFile it renames it to LastFile.

ClearLogOnStart = 1

Tells Express32 to clear the log file each time it runs.

Also each Destination printer that has been selected has its own section where the MICR alignment settings are stored.

Express32 Command Files

Standard CMD files:

Groups of Express32 commands may be called and executed from a command file using the: **CMDFile**, **CmdFile.CMD** command. See the command reference. Any number of Express32 commands may be stored in this CRLF delimited text file. Use one command sequence per line.

DO NOT use the ||CF|| delimiters. Multiple commands on one line are not supported.

Special CMD files:

Command files may also be used to create Form Overlays using the EXPRESS32 TextBox , TextBoxFont and Print@ commands. See the example INVOICE.CMD file in the EXPRESS\SAMPLES folder. This special type of command file may be uploaded using the **Overlay** or **Graphic** commands.

NOT IMPLEMENTED IN THE EXPRESS32 BETA VERSION

Print Job Rules

Print jobs generated by your application must conform to the following rules:

Print job must be in CR LF delimited format with a FormFeed at the end of each page.

OR

Print job must be in HP PCL4 or HP PCL5 format. (Output from a windows print driver or applications' own PCL driver).

All pages in your print job are expected to have a form feed (ASCII 12) at the end of each page.

Express32

If you span Express32 commands across multiple lines, Express32 will not remove CR LF's from the document. This can be confusing so try to avoid spanning an Express32 command sequence across several lines.

Very Important:

Enclose any text strings in double quotes if there is any chance that the text may include commas, asterisks, or leading spaces.

Multiple command example:

```
||CF|| OVERLAY,MYFORM.PRN * Graphic,SKATE.PCL,2.6,2.6,1 ||CF||  
      |  
      +- Multiple command  
         delimiter
```

Command Measurement Units

Arguments in brackets are optional. Commas must be maintained for placement.

Arguments whose values must contain commas and/or asterisks should be placed within double quotes.

Arguments in {curly brackets} are optional. Defaults will be used.

UNITS:

Each command has default units assigned to a numeric argument. The units may be overridden by appending a units letter to the number. For example the PRINT@ command uses Rows and Columns as default units for positioning. If you want to position the word HELLO at two inches from the top and five inches from the left just add the letter I at the end of the number.

ex: PRINT@,2I,5I,"HELLO"

Unit choices are:

All numbers may be expressed in decimal form up to 3 decimal places. i.e. 5.375R

I = Inches

R = Rows (6 rows per inch)

C = Columns (10 columns per inch)

D = Decipoints (720 decipoints per inch)

M = Millimeters (254 mm per inch)

T = Dots (300 dots per inch)

You may express any of the values above as decimal values. Ex. You may want to print at Row 3.56.

NOTE:

Arguments in braces {} are optional.

Command Reference

Search Order for Files used with Express32

Files with PCX or PCL file extensions:

If a path is included along with the file name:
Only that path will be searched for the file.

No path included:
The EXPRESS\GRAPHICS directory is searched then the EXPRESS directory is searched.

PCX files MUST be in 2 color (black and white) format (no color or grey scale)

Files with FRM or PRN file extensions:

If a path is included along with the file name:
Only that path will be searched for the file.

No path included:
The EXPRESS\FORMS directory is searched then the EXPRESS directory is searched.

All other file extensions, no path included:

The EXPRESS directory is searched.

If a file is not found in any directory searched, an error is logged in the EXPRESS.LOG file.

CMDFILE

CMDFILE, CmdFileName.CMD

CmdFileName:

Text file containing valid Express32 commands, one command sequence per line. One CmdFile command embedded in your print file can execute dozens of Express32 commands. You may change the characteristics of your print job without changing your source code.

Also see "Special Command files" in the Overlay and Graphic commands below.

Express32

OVERLAY

OVERLAY, FileName.ext

This command will be executed each time a form feed is encountered in a print job.

Use one or more times per job. You may change the Overlay in the middle of a print job with no ill effects.

FileName.Ext :

The name of the form overlay, FRM, PRN, PCX or PCL graphics file, or Express CMD file to be uploaded to the printer. This may be used to print a graphic image on any or pages. The image will be uploaded only once .

Special Command File:

FileName may also be a command text file containing BoxText, BoxTextFont, and Print@ commands used to define a form overlay. The filename MUST have a CMD extension in this special case.

Not implemented in the Beta version of Express32

Example:

```
||CF|| Overlay , MYFILE.PRN ||CF||
```

Or

```
||CF|| Overlay , MYFORM.CMD ||CF||
```

GRAPHIC

GRAPHIC, FileName.Ext, XPos, YPos, Number (1-20)

Uploads a PCX or PCL graphic file to the printer and positions it to Xpos, Ypos on the page. You may include any number of form overlays or graphic images per print job (depending on printer memory).

FileName.Ext :

The name of the form overlay, PCX or PCL graphics file, or Express CMD file to be uploaded to the printer. This may be used to print a graphic image on any or pages. The image will be uploaded only once .

PCX graphics files contain no positioning information but FRM, PCL or (PRN) print files generated by windows printer drivers and Express CMD files do. The Xpos and Ypos, below only affect files that do not contain positioning information.

Special Command File:

FileName may also be a command text file containing BoxText, BoxTextFont, and Print@ commands used to define a form overlay. The filename MUST have a CMD extension in this special case. XPos and yPos will have no effect on a CMD form overlay file.

Not implemented in the Beta version of Express32

XPos:

Default Units = Inches

This is the distance in decimal inches from the left edge of the printable area to the top left corner of the graphic.

YPos:

Default Units = Inches

Express32

This is the distance in decimal inches from the TOP of the page - or - the top of the check if the StartCheck command is in effect.

GraphicNo (1-20)

Optional.

Defines an area of printer memory to save the graphic in. If you are uploading a different graphic on each page you will want to assign a graphic number so that printer memory will be reused.

Using the same Graphic No with a different graphic file name will overwrite the previous image stored with that number.

Leave blank if you are only using a few different graphic images. Express32 will assign a graphic number automatically.

Example:

```
||CF||GRAPHIC,MYIMAGE.PCX,2,4,1||CF||
```

Prints the graphic MYIMAGE.PCX 2 inches from the left and 4 inches from the top of the page.

OR

```
||CF||GRAPHIC,MYFORM.CMD,0,0,1||CF||
```

PAGE32

PAGE32, Orientation, PageWidth, PageLength

Orientation:

Changes the orientation of the print job. Express32 can change the orientation of the print job but it is up to you to format the page to fit the new orientation. Portrait or Landscape.
Default = P

Value range: P or L

PageWidth:

Width of portrait orientation page in inches.
Default = 8.5 "

PageLength:

Length of portrait page in inches.
Default = 11"

Express32

PAGE (obsolete)

```
PAGE, Spacing, ChecksPerPage, BottomMargin{,  
LeftMargin, Orientation(P/L),  
DefaultMicrLineFormat}
```

This command is included for backwards compatibility with Express v 2.x. Do not use.

BINS

BINS, #|#|#|#|#|#|#|#

Where # is a valid paper source bin number. You may define a paper source bin for each copy of a page when using the Copies command. Each bin number is delimited by a vertical bar.

Default: 1|1|1|1|1|1|1|1|1

HP LaserJet IV:

- Bin 1 = 250 sheet paper drawer.
- Bin 4 = Flip out tray in front.
- Bin 5 = 500 sheet optional paper drawer.

HP Laserjet 5si:

- Bin 8 = Tray 1
- Bin 1 = Tray 2
- Bin 4 = Tray 3

Example:

```
||CF|| Bins, 1|4|5|1|1|1|1|1 ||CF||
```

Express32

COPIES

COPIES, NumberOfCopies

Range 1- 99

NumberOfCopies:

Number of copies that will print of current and all subsequent pages until number of copies is re set.

Example:

```
||CF||Copies,3||CF||
```

Using the above BINS example, Copy 1 will be drawn from bin 1, copy 2 from bin 4, and copy 3 from bin 5.

PRINT@

PRINT@, Row, Col, Text {,PointSize, Style, TypeFace}

Row, Col, Text {, PointSize, Style, TypeFace} Information in brackets for PCL5 type printers ONLY. If using commas or asterisks in the text please enclose the text in double quotes.

Example: "1234 Main St., Ste 101"

Row:

This is the distance in rows from the TOP of the page - or - the top of the check if the StartCheck command is in effect. There are six rows per inch.

Col:

The distance in columns from the left side of the page. There are ten columns per inch.

Note:

If you set BOTH Row and Col to zero, Text will print in the specified font at the current cursor location on the page.

Text:

The text to be printed at this position. Please enclose your text in double quotations EX: "1234 Main St. , Ste 104"

Note: The arguments that follow are only available on HP PCL5 type printers.

PointSize:

The height of the font in points. There are 72 points per inch. Default is 8 points.

Express32

Style:

Choices are N for Normal, B for Bold, I for Italic, and BI for Bold Italic. Default is Normal.

Typeface:

Choices are H for Helvetica and T for Times Roman. Or you may enter the four digit typeface number that identifies the typeface. This number can be found in your printer manual or on the typeface list that can be printed from the front panel of your laser printer. Some common numbers for the Hewlett Packard LaserJet 4 are:

| | |
|-------------|---------------|
| 4099. | Courier |
| 4113. | CG Omega |
| 4140. | Clarendon |
| Condensed | |
| 4148. | Universe |
| 4168. | Antique Olive |
| 4197. | Garamond |
| 4297. | Marigold |
| 4362. | Albertus |
| 16602. | Arial |
| 16901. | Times |
| 4102. | Letter Gothic |
| 0. | Line Printer |

NOTE:

If the typeface you are specifying is fixed pitch, the pointsize for a fixed pitch font now becomes the pitch size. The smaller the pitch the larger the font will be.

Example:

```
||CF||Print@,15,35,"Smith,Jones",12,BI,4148||CF||  
Prints in Universe 12 pt bold italic.
```

```
||CF||Print@,15,35,"Smith,Jones",6,BI,4102||CF||  
Prints in Letter Gothic 6 characters per inch, bold italic.
```

TEXTBOX

TEXTBOX, *Text*, *Width*, *Height*, *LineThickness*,
FillPattern, *xPos*, *yPos*

Text:

Text to be printed in box. (Use leading spaces to center text). Text must be in double quotes.

Width:

Width of box. Default is decimal inches. We recommend that you override the units with (C) columns. Column units are easier to work with in most cases.

Height:

Height of box. Default is decimal inches. We recommend that you override the units with (R) rows. Row units are easier to work with in most cases.

LineThickness:

Line thickness in decimal inches. A thickness of .006 or 4D (decipoints) makes a nice line.

FillPattern:

Grey dot fill. Use a value from 0 to 100 percent. Zero is no fill, 100 is solid black.

Xpos:

Location of box from left edge of paper in decimal inches. We recommend that you override the units with (C) columns. Column units are easier to work with in most cases.

Ypos:

Location of box from top of page in decimal inches. We recommend that you override the units with (R) rows. Row units are easier to work with in most cases.

Express32

To simplify the creation of form overlays using the TextBox command, you may specify the Xpos and Ypos in terms of the previous or next print positions. This concept is a little tricky but you will see that it can be very useful. You can define entire groups of boxes all aligned perfectly. You can then move these groups by changing the X and Y positions of the first box defined.

NEXT parameter:

When used in Xpos: Current box will start printing at the right edge of previous box.

When used in Ypos: Current box will start printing at bottom edge of previous box.

LAST parameter:

When used in Xpos: Current box will start printing at left edge of previous box.

When used in Ypos: Current box will start printing at top edge of previous box.

The following example is from our sample INVOICE.CMD file. The entire invoice header is defined with only absolute positioning command. The rest are printed relative to the TextBox command preceding it.

```
;Invoice header
;TextBox, Width, Height, LineThick, Fill, xPos, yPos
TextBox, "F.O.B. POINT",13c,1.5R,5D,0,0C,20R
;subsequent lines use relative positioning
TextBox, "CUSTOMER ORD NO.",16c,1.5R,5D,0,Next,Last
TextBox, "SHIP VIA",18c,1.5R,5D,0,Next,Last
TextBox, "TERMS",23c,1.5R,5D,0,Next,Last
TextBox, "OUR ORD NO.",10c,1.5R,5D,0,Next,Last
TextBox, "",13c,2R,5D,10,0C,Next
TextBox, "",16c,2R,5D,10,Next,Last
TextBox, "",18c,2R,5D,10,Next,Last
TextBox, "",23c,2R,5D,10,Next,Last
TextBox, "",10c,2R,5D,10,Next,Last
```

Express32

Note that the entire group of boxes can be moved by simply changing the X and Y positions of the F.O.B. POINT text box.

Express32

TEXTBOXFONT

TEXTBOXFONT, PointSize, Style, TypeFace

Sets the font for all subsequent TextBox commands

PointSize:

Height of font in points

Style:

Normal, Bold, Italic, BoldItalic

N, B, I, BI

TypeFace:

See description in Print@ command.

PCL

PCL, EscSequence

Embeds a PCL escape sequence directly in to the print job. Use the caret (^) for the Esc character. Intimate knowledge of the PCL language is required to use this command.

There are times when you may want to send your own PCL escapes to the printer. This command facilitates the use of embedded PCL escapes.

EscSequence

Some applications make it difficult to embed straight PCL escape sequences because the escape character cannot be entered from the keyboard. This command lets you send the ESC character by using the caret (^) character.

Please refer to the HP PCL technical reference documentation for more information on how to use PCL escape sequences.

Example:

```
||CF|| PCL, ^(579L^(s1p12v0s0b31402T ||CF||
```

Sets the current font to 12 pt Wingding on a HPIV printer. Be careful! Strange things can happen if you do not fully understand the PCL language.

Express32

CODE39

CODE39, Text, XPos, YPos, PointSize, NarrowBarWidth

Prints a CODE 3 of 9 barcode at the specified position. Most CODE39 barcode text is surrounded by asterisks. For safety please enclose the CODE39 text in double quotes.

Setting the height to a negative number will prevent the text from printing below the barcode.

Text:

This is the data that is to be encoded as a Code 3 of 9 barcode. Usually started and ended with an asterisk.

XPos:

Units = Decimal Inches

The horizontal position of the object from the left hand side of the page.

YPos:

Units = Decimal Inches

The vertical position of the object from the top edge of the page OR if used between STARTCHECK and ENDCHECK it is the distance from the top of the current logical check to the object.

Height:

Units = Points

Height in points of the barcode. Certain minimum barcode standards will prevent you from printing a barcode smaller than about 18 points high (depending on the number of characters encoded).

NarrowBarWidth

Units = Decipoints

You can adjust the width of the barcode by changing the narrow bar width. The default is 6 decipoints (the minimum allowed).

Example:

```
||CF||Code39,"*123456AZD*",3.5,4.67,36,10||CF||
```

Express32

STARTCHECK

STARTCHECK

No Arguments. Causes express to go in to logical page mode where all vertical positions are calculated relative to the top of the current logical check.

This command is used to define the beginning of a new check on the page. The print position of the check is incremented to the next position on the page OR to the first one on the next page. Remember — you must provide the actual page breaks in your print data.

Example:

```
||CF||STARTCHECK||CF||
```

MICRPAGE

`MICRPAGE, ChecksPerPage {, Spacing, BottomMargin
,DefaultMicrLineFormat, DefaultCheckNoFormat }`

ChecksPerPage:

This is the number of checks that will print on each page. This number and the SPACING define how many and where the checks will print on each page.

NOTE:

It is up to you to insert page breaks in your print data. Express32 makes no provision to break pages if the number of checks per page exceeds the number entered here.

Spacing:

Default Units = Inches

Sets the spacing in inches between the top of the paper and the bottom of the first check on the page. The bottom of the check is defined as the edge of the paper immediately below the MICR line (approx. 1/8" below the MICR line). This is where the paper is perforated unless you are printing the check at the bottom of the page.

Some examples for a standard business check.

Spacing = 3.5

3 checks per page.

Spacing = 3.5

1 check per page and the check is located at the top.

Spacing = 7

1 check per page and the check is in the middle

Spacing = 11

Express32

1 check per page and the check is at the bottom of the page.

BottomMargin:

Default Units = Inches

BottomMargin is used when you define a check with a stub directly below it. An example would be printing 2 checks per page, each with a stub below that is 2 inches high. You would define MICRPAGE as follows:

ChecksPerPage = 2

Spacing = 3.5

BottomMargin = 2

If the check does not have a voucher (stub) below it then BottomMargin is always zero.

DefaultMicrLineFormat:

Determines where the MICR encoded check number is located and how it is formatted.

B:

Use B for 8.5" business checks. The check number occupies 8 positions and is located to the left of the Transit number.

P1:

For 6" personal checks. This format will print the check number after the Transit number with one space between the check number and the OnUs number..

P2:

For 6" personal checks. This format will print the check number to the RIGHT of the OnUs number.

Example:

```
||CF||MICRPAGE, 3, 3.5, 0, B, #####||CF||
```

Three checks per page, each check is 3.5 inches high, no bottom stub, Business Micr format, eight digits allowed for the check number.

Defaults:

ChecksPerPage = 1

Spacing = 11" (Check at bottom of page)

BottomMargin = 0

MicrLineFormat = B

CheckNoFormat = #####

Express32

SIGLINE

SIGLINE, SigLineNo {, WordsBelow,
SignatureGraphicFile, VOffset }

Up to 3 signature lines may be printed on your checks. Signature graphics may also be printed as needed.

SigLineNo:

Use numbers 1,2,3. Signature line 1 prints closest to the MICR line, 2 prints above it, and 3 prints above 2.

WordsBelow:

Words to be printed below the signature line. Example: COMPROLLER.

SignatureGraphicFile:

This is the PCX or PCL file name for the signature graphic file. The file is expected to be located in the Express32 directory unless a path is specified.

VOffset:

Default Units = Decipoints

Sometimes the signature graphics need to be moved up or down a little. Use a more negative value to move up and a more positive value to move it down. Units are in decipoints. 720 decipoints to an inch. Try 90 as a starting point.

Example:

```
||CF||SIGLINE, 1, "Words Below line 1", MYSIG.PCX, 90||CF||
```

MICRAMOUNT

MICRAMOUNT, CheckDollarAmount

CheckDollarAmount:

Decimal dollar amount of the current check being printed.

MICR encodes the dollar amount of the check in the lower right corner of the check. DollarAmount is a numeric decimal number. Non numeric characters will be stripped. Number will always be encoded with two decimal places. The decimal will not be printed.

WARNING! Check with the bank that the account is drawn on for permission to encode this field. Not all banks will allow you to encode this field.

Example:

||CF||MICRAMOUNT,995.45||CF||

Express32

MICRLINE

`MICRLINE, CheckNo, TransitNo, OnUsNo {,
MicrLineFormatOverride, CheckNoMaskOverride}`

CheckNo:

This is the sequence number for the current check. For business checks you are limited to 8 characters and 4 characters for personal checks.

TransitNumber:

This is sometimes called the routing number. This number is used by the banking system to identify the bank on which the check was drawn.

This is part one of the two part account number that makes up the MICR line. The number is always 9 digits long. Sometimes there is a Dash character () inside it. It is bounded on each side by the Transit character. Do not enter the transit characters here, they are supplied by Express32.

OnUsNumber:

This is part two of the MICR line. This identifies the account on which the check is drawn. The number may contain spaces, Dash Characters and the OnUs Character.

MicrLineFormat:

The Micr line normally defaults to the format defined in the MICRPAGE command. You may override the format here. Leaving this item blank results in the default format. See DefaultMicrLineFormat in the MICRPAGE command for details.

CheckNoMask:

This overrides the format of the check number as defined in the DefaultCheckNoMask in the PAGE command. Leaving this item blank results in the default format defined in the PAGE command.

This field determines the format of the check number field in the MICR line. The format may be made up of numbers (0123456789), underscores (_) and pound signs (#).

The underscore characters are replaced by spaces.

The actual check number is right justified in the positions defined by the #'s. Unused positions are filled with zeros.

Example:

`_3200####_` would yield: " 32000999 " if the check number was 999.

SEE DEFAULT FORMATS IN THE PAGE COMMAND.

Example:

```
||CF|MICRLINE,100099,123456789,123456789=,#####|CF||
```

Express32

ENDCHECK

ENDCHECK

EndCheck tells Express32 that the current check is defined and it now time to print it on the page at the next available print position. Express32 resumes calculating vertical positions from the top of the page.

Example:

```
||CF||ENDCHECK||CF||
```

MICR Symbol Translations

NOTE:

Please see APPENDIX A for more information about MICR check encoding.

Since the MICR symbols are not part of the IBM PC character set we must use substitute symbols to identify the E13B MICR symbols. They are identified below.

Numeric Symbols:

1 2 3 4 5 6 7 8 9 0

: Transit Symbol (:) colon

= OnUs Symbol (=) equal

< Dash Symbol (<) left angle bracket

; Amount Symbol (;) semicolon

Space (. _) period or underscore

Under normal circumstances you will only need to use the Dash, OnUs and Space characters when defining the MICR line.

Other MICR fonts:

We support MICR fonts resident in several MICR laser printers and font cartridges. Please call our tech support dept .for information on using other vendors fonts.

Command Order

The following Express32 commands must be used between the StartCheck and EndCheck commands:

MicrAmount

MicrLine

SigLine

When other commands using positioning coordinates are used between the STARTCHECK and ENDCHECK command, the print positions are calculated relative to the top left corner of the current logical check. When used outside of the STARTCHECK/ENDCHECK commands, positions are calculated relative to the top left corner of the physical page.

Debugging

The best way to debug is to print your data without being processed by Express32. Look at your printed output to make sure your commands are correct.

Page breaks will not be affected by Express32. If your job is not page breaking properly please check your code that is producing it.

The easiest way to embed the commands is to locate them at the end of your lines of text in your output document. This way if you need to change a command format it will not cause any spacing problems with your actual data.

If there are any errors they will be printed to the EXPRESS.LOG file in your local workstations' WINDOWS directory. The log file may be erased each time Express32 is executed. See the Log File section.

The only time Express32 will have a problem seeing your commands is if your program prints Express32 commands in a non printer resident font.

Also Express32 will not work if your application sends its output to the laser in graphics mode as many Windows applications do. If this is the case then select the Generic ASCII text printer driver in Windows to print your data.

Final Notes:

Microsoft Windows v 3.1

Express32 is a 32 bit application and will not run under Windows 3.1

APPENDIX A

A word about printing MICR documents:

What is a MICR check?

A MICR check is simply a negotiable banking instrument. In this case the check is printed in its' entirety on a laser printer. MICR is an acronym for "Magnetic Ink Character Recognition". The MICR portion of a check is line of odd looking characters about one fourth of an inch from the bottom of the check. The MICR characters are called the E-13B character set as defined by the banking industry.

Why is the MICR line so important?

When a check is processed by the banking system it is subjected to being read by up to 30 high speed reading machines. The placement and alignment of the MICR line is absolutely critical if the check is to be read accurately by these machines. Also the shape and density of the MICR characters must be up to the stringent standards set down by the banking industry.

Misreads by the scanning equipment cost the banking industry time and money which eventually filters down to people who actually write the checks. Misread checks can be identified by looking through your checks when they come back from the bank. Generally a bank will attach a thin strip of paper to the bottom of a returned check with your MICR account numbers printed on it. This indicates that the check could not be read by the high speed sorters. Please note that misreads may be caused by a variety of mechanical problems and are not always caused by the printing of the MICR number.

The design of Check Factory Express32 incorporates the stringent specifications as laid down by the banking industry. Following these instructions will assure adherence to MICR printing standards.

Transit Number:

This field is located to the right of the check number field on 8 1/2" business checks and is the leftmost field on 6" personal checks. This is a nine digit field usually made up of nine numeric characters and sometimes an E13B Dash symbol. This field on the check is always surrounded on both sides by a transit character.

On Us Number:

This field is always directly to the right of the transit field and may contain up to 18 numeric and symbol characters. The only symbols used in this field are the Dash symbol and the OnUs symbol. The field is usually terminated by the OnUs symbol but there may also be more numbers after the symbol. The format is determined by the issuing bank.

VERY IMPORTANT NOTE!

Most 6" personal checks have the check sequence number embedded in the OnUs field. This can be very confusing when you are trying to determine which numbers make up the OnUs field. Do not include the check sequence number in the OnUs number on a business format check! If you have any question as to what the actual OnUs number is please call the bank that the check is written on. Do not call Trans-Micro, we will not be able to help you with this problem.

**** Special Condition ****

There is normally a space between the Transit Field and the OnUs field. Some banks may require that a special code number be placed in this position. Please be careful when designing the format for the MICR line.

Additional Information:

MICR Toner cartridges for many laser printers are available from Trans-Micro. Please call our sales office form more information. (800) 749-2335.

High Quality blank check stock can be purchased from Trans-Micro at (800) 749-2335

Information on MICR check printing standards can be obtained from the American Bankers Association in Washington, DC. Ask for information regarding ANSI Spec X-9 for the Non Impact Printing of Magnetic Documents.