Ithaca POS (Point of Sale)

Device Driver Tutorial and Overview.

Disclaimer

Information in this publication is subject to change without notice and does not represent a commitment. However, as product improvements become available, we will make every effort to provide updated information for the products described in this publication.

Copyright

Trademarks

PcOS is a registered trademark of. TransAct. IBM is a registered trademark of the International Business Machines Corporation. Epson is a registered trademark of Seiko Epson Corporation..

Contents

Section 1 (General Driver Information)

Section 2 (Windows Printer Driver)

Section 3 (OPOS Drivers)

Section 4 (POSPrinter OCX, ActiveX)

Section 1 (General Driver Information)

Definitions:

OS - Operating System.

Types of OS's are: Windows (9x/Me/4.0/2000), Linux, Unix, OS2, ...

Application - A software program that a person uses to do something (i.e. a POS application)

Program - Program and Application are used interchangeably.

Driver - software that makes hardware do something (i.e. something useful, we hope). A driver translates (or converts) a software command to a command that specific hardware can understand.

Types of drivers: Printer driver, Port driver, OPOS driver, USB driver...

OCX/ActiveX - a software component that utilizes Microsoft's OLE (Object linking and embedding).

API - Application Programming Interface. In the generic sense "The API" refers to the Windows Operating system API.

Types of API's: Windows API, Visual Basic API, Delphi API, MFC API

Windows Printer Driver:

A Windows printer driver is a Microsoft specific, Microsoft defined, type of driver that the OS uses to translate drawing commands by a Window's application to a specific printer's command set. A window's printer driver is a graphics page mode driver. It is not a POS (Point of Sale) driver. (Have you ever had to open a cash drawer that was connected to your printer at home?)

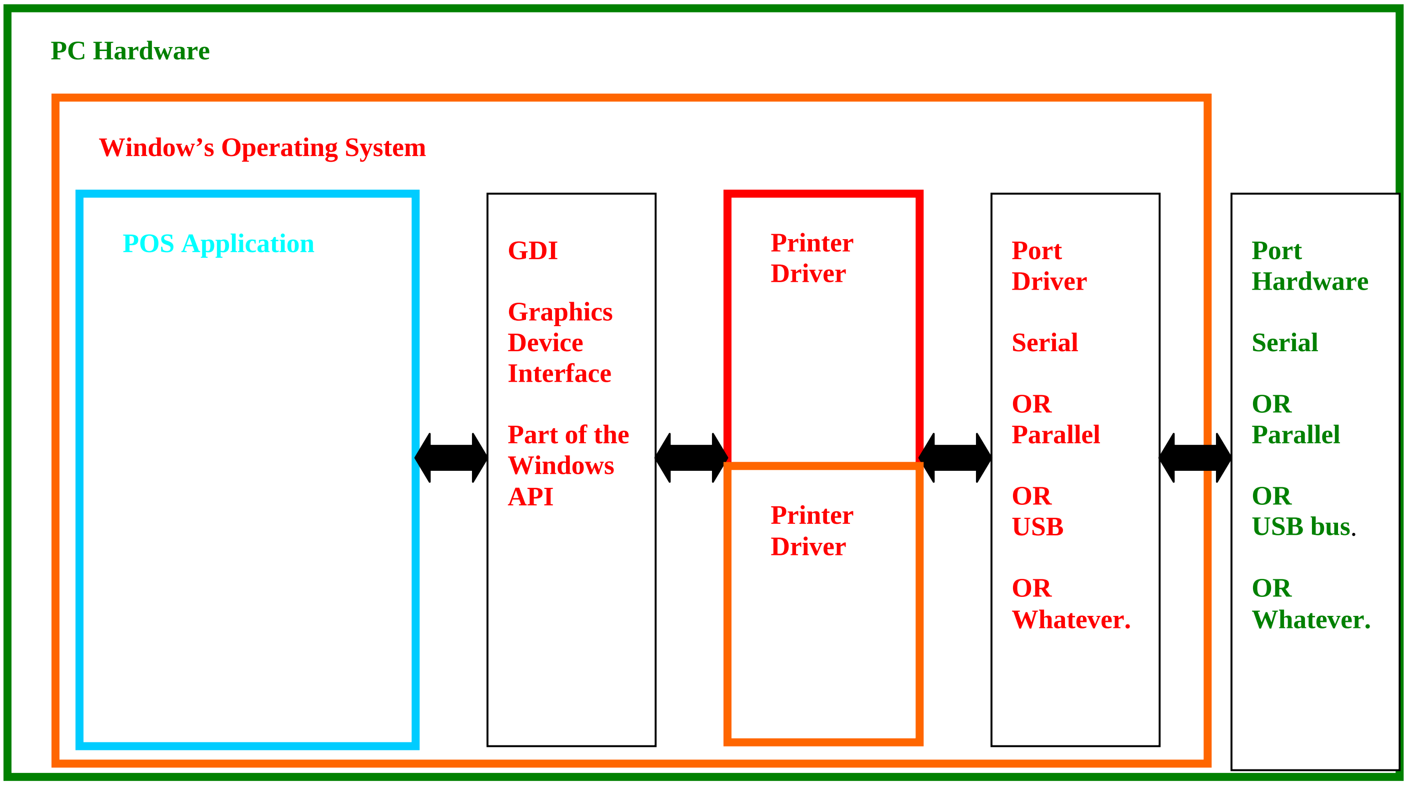


Fig A

When to use a printer driver:

Use a printer driver when writing a program that uses the Window's API to send print information to the printer.

When not to use a printer driver:

When a program wants to send printer command codes to the printer or when a program wants to get information back from a printer.

OPOS driver

An OPOS driver is not a printer driver.

An OPOS driver is a POS device driver that an application uses to get a POS device to do something. Types of OPOS drivers are POSPrinter, Scanner, Cash drawer, Pole Display, MICR, Scale, ...

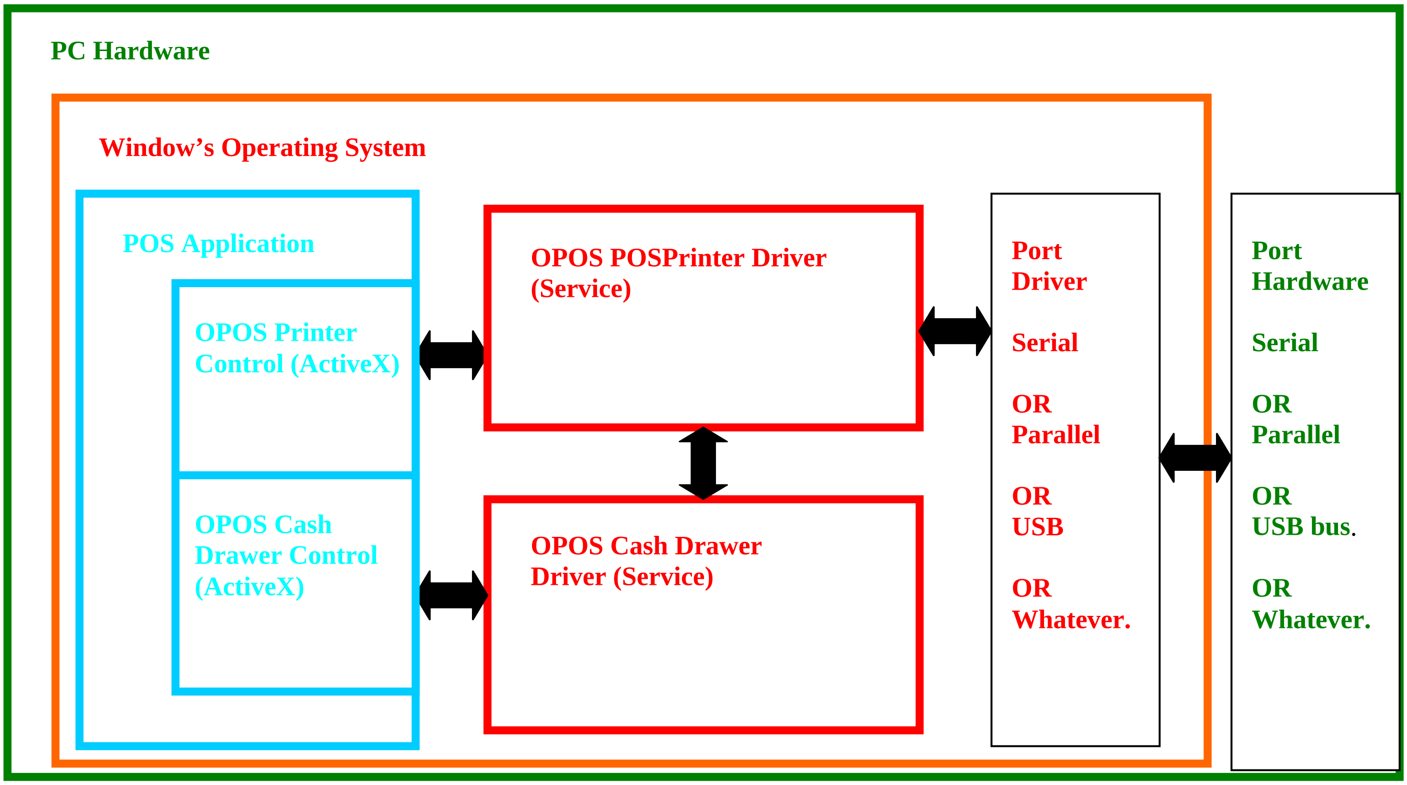

PC Hardware
Window’s Operating System
POS Application
OPOS Printer Control (ActiveX)
OPOS Cash Drawer Control (ActiveX)
OPOS POSPrinter Driver (Service)
OPOS Cash Drawer Driver (Service)
Port Driver
Serial
OR Parallel
OR USB
OR Whatever.
Port Hardware
Serial
OR Parallel
OR USB bus.
OR Whatever.

Fig B

When to use an OPOS driver:

When you want to be able to access all the features of the POS device and you do not want to send printer/device specific commands to the printer/device. When you want to get information back from the device in a standard (OPOS) way. When you want to be able to use devices from different vendors.

When not to use an OPOS driver:

When you want to send device specific commands to the device/printer.

POSPrinter ActiveX Control (OCX)

This is not a driver; it is a software component that provides a connection from an application to a port driver. This allows an application to communicate "directly" with a printer. This approach allows an application to send the commands, it wants to, to the printer. This approach is similar to the "old DOS days" of talking to a POS printer.

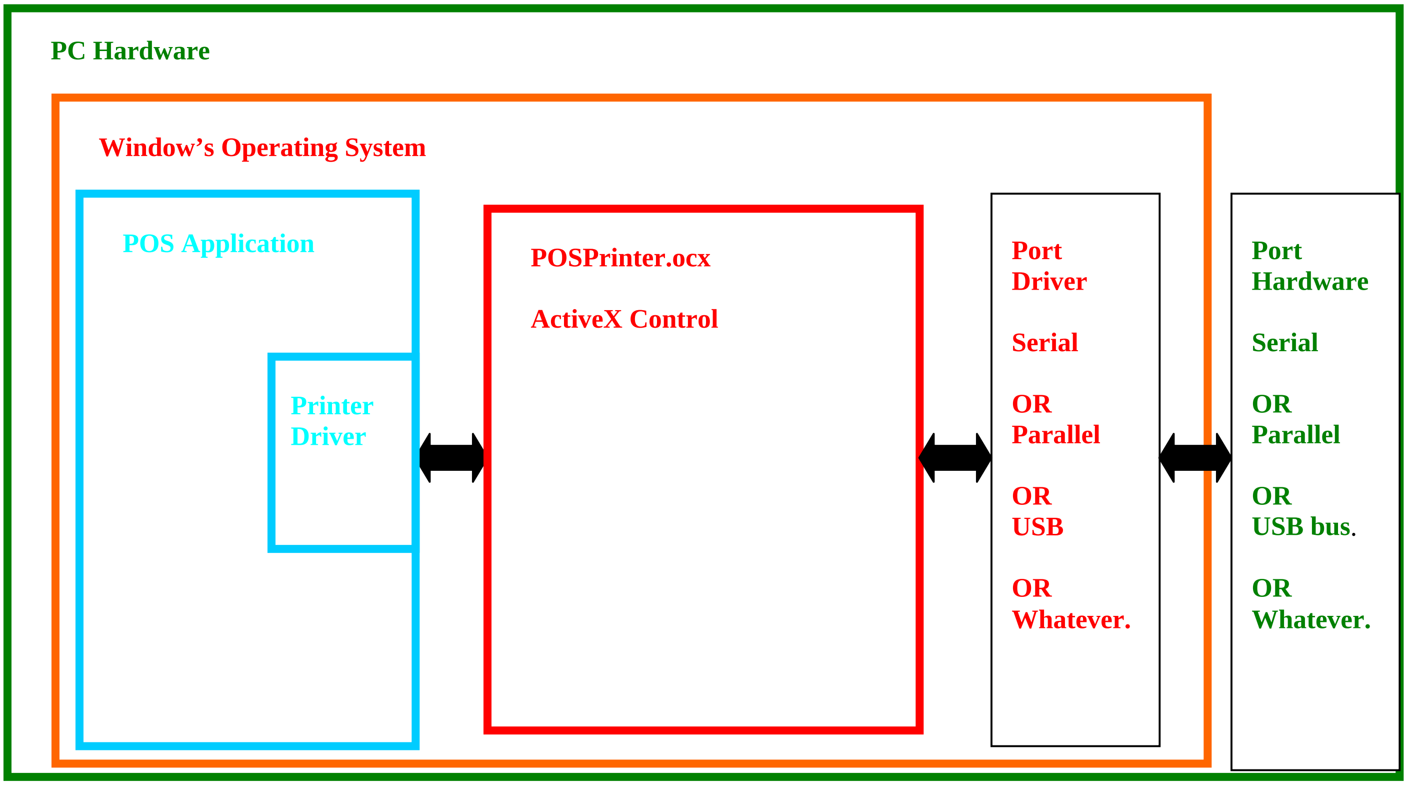


Fig C

When to use the POSPrinter OCX:

When a customer (application writer) wants **control** of what is sent to the printer. The customer must take the responsibility of sending the correct codes to the printer as well as detecting error conditions with the printer and graceful recovery from error situations.

When not to use the POSPrinter OCX:

When the customer does not want to deal with the low-level commands sent to the printer.

QUIZ (General Driver Info.)

What is a driver?

Software that makes hardware do something. A driver translates (or converts) a software command to a command that specific hardware can understand.

Why should you not use a Windows Printer driver to send command codes from an application to the printer?

a) A Windows Printer driver was intended to shield the application from the specifics of a printer.
b) There are too many OS layers out side the printer driver's control.
c) Doesn't always work.
d) All of the above.

Why do OPOS drivers exist?

a) To add confusion to my life.
b) To provide a standard interface to POS devices.
c) So POS application writers can use devices from different vendors.
d) Both b) and c).
e) All of the above.

Section 2 (Windows Printer Driver)

Installing a Windows printer driver

Go to the printer's folder. To do this, click "Start" menu, then "Settings", and click "Printers".

Double click on the "Add Printer" Icon, follow the steps a presented to you. When the "Have Disk" button appears, click it and, insert the floppy disk with our drivers on it. Continue to follow the steps as present by the "Add Printer" wizard. If you have a question see the help that comes with the operating system (i.e. "printers" topic).

Configuring the printer driver.

The standard Window's printer driver configuration properties are described in the operating system's help under the "printers" topic. The Ithaca specific configuration is described below.

Start/EndDoc Tab

Uses these setting to cause additional actions to happen at the start and end of a document.

Note: For these options to work the printer driver must be in "bi-directional" mode. To check this, go to "Details" tab, "Spool settings". For NT and Win 2000, go to the "ports" tab.

Additional StartDoc Commands: This is where you place text and binary commands that you want sent to the printer before a document is printed.

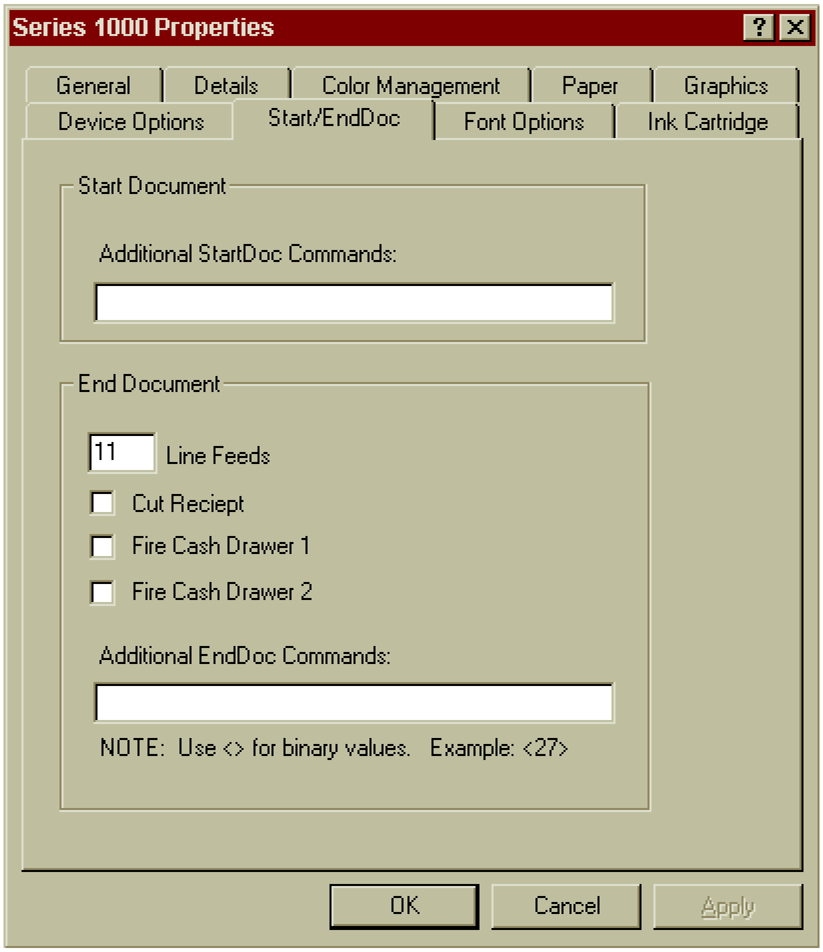
Additional EndDoc Commands: This is where you place text and binary commands that you want sent to the printer after a document is printed.

Line Feeds: This is where you enter the number of line feeds that will be done after a document is printed.

Cut Receipt: This option will send a cut command to the printer after the document is printed.

Fire Cash drawer 1: This option will send a fire cash drawer 1 command to the printer after the document is printed.

Fire Cash drawer 2: This option will send a fire cash drawer 2 command to the printer after the document is printed.



Additional StartDoc Commands:

11 Line Feeds

□ Cut Reciept

□ Fire Cash Drawer 1

□ Fire Cash Drawer 2

Additional EndDoc Commands:

NOTE: Use <> for binary values. Example: <27>

Font Options Tab (Win 9x/Me only)

These options control how font processing is handled in the printer driver.

Note: Selecting "Always map fonts to device resident fonts" causes the printer driver to behave in a way that is not how the operating system expects a printer driver to operate. This is not the default mode of operation and can cause the printed output to vary a lot depending on the settings you choose.

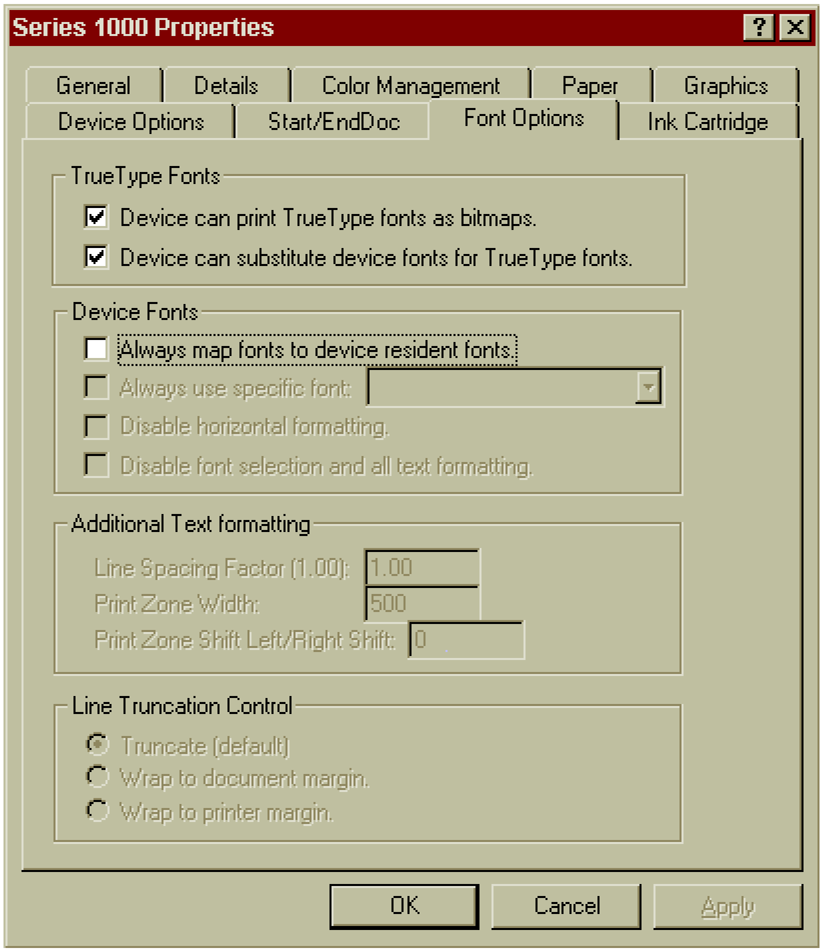
True Type Fonts These settings control capability flags that are reported to the operating system.

Device Fonts: These settings control how printer resident fonts get selected. These settings force the driver to behave in a non-standard way.

These "Device Font" settings can make up for an application that does not select a printer resident font.

Additional Text formatting These settings control how text gets printed in the print zone and the spacing between lines.

Line Truncation Control These settings control how a line that is too long to fit is handled.



TrueType Fonts

☑ Device can print TrueType fonts as bitmaps.

☑ Device can substitute device fonts for TrueType fonts.

Device Fonts

□ Always map fonts to device resident fonts.

□ Always use specific font:

□ Disable horizontal formatting.

□ Disable font selection and all text formatting.

Additional Text formatting

Line Spacing Factor (1.00): 1.00

Print Zone Width: 500

Print Zone Shift Left/Right Shift: 0

Line Truncation Control

● Truncate (default)

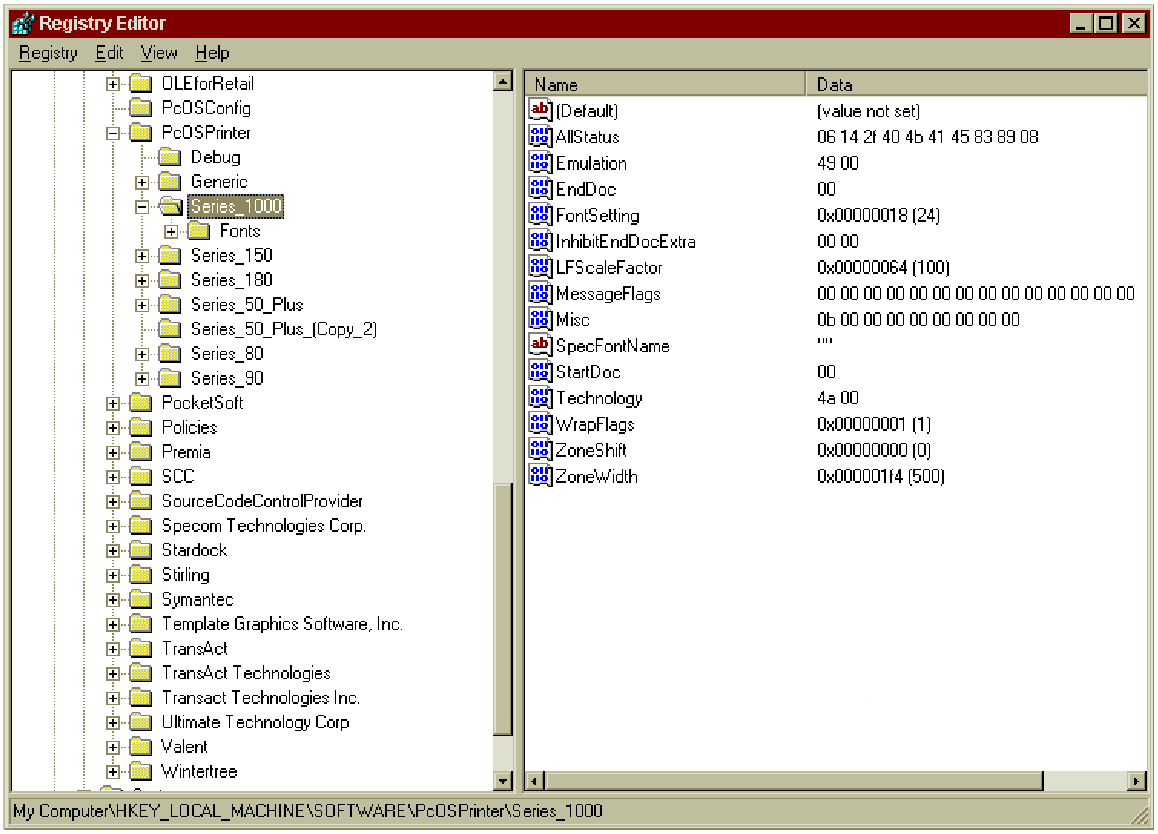
○ Wrap to document margin.

○ Wrap to printer margin.

Note: The custom printer driver settings are stored in the Window's system registry. These custom settings persist even if you uninstall the driver. You can view, edit, and delete these setting via RegEdit.exe that comes with Windows. **Using RegEdit.exe can render you system unusable and lead to a loss of data on your computer, extreme caution should be used when editing the system registry**. You should backup you registry before you edit it (see Window's help for more information).

It is recommended that the custom settings be changed via the printer's property pages as discussed previously in this document.

To view, edit, or delete the custom settings, start RegEdit.exe and go to HKEY_LOCAL_MACHINE\SOFTWARE\PcOSPrinter\Series_1000 for the Series 1000 printers.



Name	Data
(Default)	(value not set)
AllStatus	06 14 2f 40 4b 41 45 83 89 08
Emulation	49 00
EndDoc	00
FontSetting	0x00000018 (24)
InhibitEndDocExtra	00 00
LFScaleFactor	0x00000064 (100)
MessageFlags	00 00 00 00 00 00 00 00 00 00 00 00 00 00
Misc	0b 00 00 00 00 00 00 00 00
SpecFontName	""
StartDoc	00
Technology	4a 00
WrapFlags	0x00000001 (1)
ZoneShift	0x00000000 (0)
ZoneWidth	0x000001f4 (500)

My Computer\HKEY_LOCAL_MACHINE\SOFTWARE\PcOSPrinter\Series_1000

Printing via a printer driver (Printer Font)

See the documentation that comes with Visual Basic for detailed information on all the function calls that can be made to print via the Windows Printer drivers.

```
'SELECT OUR PRINTER BY NAME
For Each x In Printers
    If x.DeviceName Like "Series 1000" Then
        Set Printer = x
        Exit For
    End If
Next


'SELECT FONT BY NAME
Printer.FontName = "NLQ 10cpi"
Printer.Print "     20 Bomax Rd"
Printer.Print "   Ithaca, NY 14850"

Printer.Print ""
Printer.Print ""
Printer.Print ""

Printer.FontName = "Draft 16cpi"
'SELECT BOLD PRINT
Printer.Font.Bold = True
Printer.Print "ITEM                            PRICE"
Printer.Font.Bold = False

Printer.Print "Cheese Wiz                      $1.49"
Printer.Print "Crackers                        $2.59"
Printer.Print "Laundry Soap                    $3.29"
Printer.Print "Printer Paper                   $3.89"
Printer.Print "Rubber Bands                    $ .99"
Printer.Print "Soda                            $1.49"
Printer.Print "Candy                           $2.29"
Printer.Print "-------------------------------------"
Printer.Print "          Sub Total            $16.03"
Printer.Print "          Tax                   $1.28"
Printer.Font.Bold = True
Printer.Print "          TOTAL                $17.31"
Printer.Font.Bold = False
Printer.Print "          ---------------------------"
Printer.Print "          Cash                 $20.00"
Printer.Print ""
Printer.Font.Bold = True

'CHANGE COLOR
Printer.ForeColor = RGB(255, 0, 0)
Printer.Print "          Change Due            $2.69"
Printer.ForeColor = RGB(0, 0, 0)
Printer.Font.Bold = False

Printer.Print ""
Printer.Print ""
Printer.Print ""
Printer.Print ""

Printer.EndDoc
```

Printing via a printer driver (System/Graphical/TrueType Font)

This is the preferred way to print via a Windows printer driver to a page printer.

```
'SELECT OUR PRINTER BY NAME
For Each x In Printers
   If x.DeviceName Like "Series 1000" Then
      Set Printer = x
      Exit For
   End If
Next


'SET FONT SIZE
Printer.FontSize = 10

'SELECT FONT BY NAME
Printer.FontName = "Arial Black"
Printer.CurrentX = 1000
Printer.Print "20 Bomax Rd"
Printer.CurrentX = 800
Printer.Print "Ithaca, NY 14850"



Printer.Print ""
Printer.Print ""
Printer.Print ""

'SELECT FONT BY NAME
Printer.FontName = "Arial"

'SELECT BOLD PRINT
Printer.Font.Bold = True

'SET PRINT POSITION
LinePos = Printer.CurrentY
Printer.Print "ITEM"
Printer.CurrentX = 2800
Printer.CurrentY = LinePos
Printer.Print "PRICE"
Printer.Font.Bold = False

LinePos = Printer.CurrentY
Printer.Print "Cheese Wiz"
Printer.CurrentX = 2900
Printer.CurrentY = LinePos
Printer.Print "$1.49"

LinePos = Printer.CurrentY
Printer.Print "Crackers"
Printer.CurrentX = 2900
Printer.CurrentY = LinePos
Printer.Print "$2.59"

LinePos = Printer.CurrentY
Printer.Print "Laundry Soap"
Printer.CurrentX = 2900
Printer.CurrentY = LinePos
Printer.Print "$3.29"
```

```
LinePos = Printer.CurrentY
Printer.Print "Printer Paper"
Printer.CurrentX = 2900
Printer.CurrentY = LinePos
Printer.Print "$3.89"

LinePos = Printer.CurrentY
Printer.Print "Rubber Bands"
Printer.CurrentX = 2900
Printer.CurrentY = LinePos
Printer.Print "$  .99"

LinePos = Printer.CurrentY
Printer.Print "Soda"
Printer.CurrentX = 2900
Printer.CurrentY = LinePos
Printer.Print "$1.49"

LinePos = Printer.CurrentY
Printer.Print "Candy"
Printer.CurrentX = 2900
Printer.CurrentY = LinePos
Printer.Print "$2.29"

Printer.CurrentY = Printer.CurrentY + 100
Printer.Line (0, Printer.CurrentY)-(3600, Printer.CurrentY)
Printer.CurrentY = Printer.CurrentY + 100


LinePos = Printer.CurrentY
Printer.CurrentX = 1300
Printer.Print "Sub Total"
Printer.CurrentX = 2800
Printer.CurrentY = LinePos
Printer.Print "$16.03"

LinePos = Printer.CurrentY
Printer.CurrentX = 1300
Printer.Print "Tax"
Printer.CurrentX = 2900
Printer.CurrentY = LinePos
Printer.Print "$1.28"

Printer.Font.Bold = True

LinePos = Printer.CurrentY
Printer.CurrentX = 1300
Printer.Print "TOTAL"
Printer.CurrentX = 2800
Printer.CurrentY = LinePos
Printer.Print "$17.31"

Printer.Font.Bold = False

'Printer.Print "---------------------------"

Printer.CurrentY = Printer.CurrentY + 100
Printer.Line (1300, Printer.CurrentY)-(3600, Printer.CurrentY)
Printer.CurrentY = Printer.CurrentY + 100
```

```
LinePos = Printer.CurrentY
Printer.CurrentX = 1300
Printer.Print "Cash"
Printer.CurrentX = 2800
Printer.CurrentY = LinePos
Printer.Print "$20.00"

Printer.Print ""

Printer.Font.Bold = True

'CHANGE COLOR
Printer.ForeColor = RGB(255, 0, 0)
LinePos = Printer.CurrentY
Printer.CurrentX = 1000
Printer.Print "Change Due"
Printer.CurrentX = 2900
Printer.CurrentY = LinePos
Printer.Print "$2.69"
Printer.ForeColor = RGB(0, 0, 0)
Printer.Font.Bold = False

Printer.Print ""
Printer.Print ""



Printer.EndDoc
```

QUIZ (Windows Printer Driver)

What is the purpose of a Windows printer driver?

a) Translate graphics commands to commands the printer can understand.
b) To be a pain in the neck.
c) Intended to shield the application from the specifics of a printer.
d) All of the above.
e) None of the above.
f) Both a) and c).

How do you select a font to print with via a printer driver?

a) Cross your fingers and hope it comes out good.
b) Send the correct escape code.
c) Do a Printer.FontName = "Draft 16cpi"
d) None of the above.

How do you turn on bold print?

a) Pray
b) Send the Escape code.
c) Do a Printer.Font.Bold = True
d) All of the above.
e) None of the above.

How do you set the print position with via a printer driver?

a) Space pad the text so it looks good.
b) Send line space and horizontal escape codes.
c) Do a Printer.CurrentY = ?? and a Printer.CurrentX = ??
d) a) (with a printer font), and c)
f) All of the above.
g) None of the above.

How do you control the color of text that gets printed.

a) Send a color escape command.
b) Do a Printer.ForeColor = RGB(255, 0, 0) for red.
c) Let someone else to it.

Section 3 (OPOS Drivers)

Installing Ithaca OPOS drivers.

To start the install process, run the SetupOPOS.exe file. Follow the on screen instructions. When you get to the "OPOS printer service object settings" follow the directions in the OPOS configuration section below.

Ithaca OPOS drivers Configuration.

To configure the OPOS drivers, either run the "Ithaca OPOS Setup" in the control panel or run the configuration via the "Start" menu, "Programs", "Ithaca OPOS", "Ithaca OPOS Setup".

You must choose the communications port (and the port setting in the case of a serial port).

At the OPOS printer configuration you can set the following.

Printer driver

Choose the printer that you are using. "SeriesAutoDetect" can automatically detect which printer is connected. This auto detection will allow switching of Ithaca series printers without changing host system settings.

Rename: This will allow you to rename the OPOS device name. Some applications look for a specific name for the device to use. This renaming will accommodate an application of this type.

Remove: This will allow you to remove an OPOS device from the system.

Set as Only: This will allow you to set this device as the only device of this type (printer). Some applications look for the first device name for the device to use. To ensure that a device is the first one, use this setup feature. Caution should be exercised when using this setup feature, this "Set as Only" will remove all other OPOS devices of this type (printer).
Note: An entry may be first alphabetically, but not first to be enumerated by the operating system.

Printing

Default Characters Per Line: This is the default number of character per print line. The number of characters per line is typically set by the application. If the application does not set the number of characters per line, this default will be used.

Line Feed Scale Factor: This controls the base/default line height/spacing. Increase this value for a larger line spacing.

Fast Bitmap Printing: Set to have bitmaps print in the fastest mode possible (while xxxLetterQuality is false). If xxxLetterQuality is set to true, the "Fast Bitmap Printing" option is overridden and bitmaps are printed in high quality mode. If "Fast Bitmap Printing" is off and xxxLetterQuality is set to false, bitmaps are printed in a medium quality mode.

Double High = Double Line Feed: If set, when a line feed is encountered after double high print is requested, a double line feed will be done.

Print Red as Bold: If set, a request for alternate color print (red) will be printed as bold.

Truncate Trailing Spaces: If set, spaces on the end of a line of text will be removed. This truncation can be used to prevent unwanted line wrapping.

General Settings

Paper Low = Paper Out: If set, paper low will not be reported until paper is out.

Inhibit Events: If set, events will not be generated.

Error Events Asynchronous Mode Only: Set to enable firing error events while in Asynchronous mode only. If not set, error events will be fired in both Asynchronous and Synchronous modes.

Set Error String on Synchronous Error: If set, the ErrorString will be set when an error occurs while calling a method in synchronous mode. Otherwise it will only be set in asynchronous mode.

Synchronous Time Out: Time in milliseconds a synchronous print method will wait before returning a OPOS_E_TIMEOUT. If set to -1 (0xffffffff), a synchronous print method will never time out (default).

Synchronous Error Wait: Time in milliseconds a synchronous print method will wait for an error condition to be removed before returning an error. If the error condition is removed before this time has elapsed, the print method will be processed.

Performance

Background Status Updating: If set, printer status will be updated in the background. This background updating will increase performance. This performance increase is because when a print request is made the status is obtained from a previous background read and not when the print method is called. This is especially important using a parallel interface, being that the parallel interface operates in half duplex.

Printer Never Busy: If set, the printer service object (“driver”) will report that the printer is not/never busy. This will greatly increase synchronous printing performance. Synchronous print calls will not wait for the printer to actually be idle. It should be noted that this “never busy” mode is not strictly compliant with the OPOS specification. This performance increase is especially important using a parallel interface, being that the parallel interface operates in half duplex.

Communications Port:

Port Name: This is the name of the communications port (ie. COM2, LPT1).

Setup Comm Port: This will display a communications setup dialog. You will be able to set things such as baud rate and handshaking. Note: these settings must match the printer you are using.

I/O Time Out Factor: This controls how long the driver waits for I/O. Its value can range from 10 to 2000. The smaller this value is, the more responsive the application will be when the printer is off or disconnected.

Paper Out:

Ignore Paper Out: If this is selected, the print functions will still attempt to print to the printer even when paper out is sensed. This functionality is provided to maximize paper use on the Series 90 printer that reports paper out as a function of the number of print lines since paper low is sensed. Use on printers other than the Series 90 is not recommended. You must have printer configuration “Paper Out Mode” set to “Disabled” to use this functionality.

Do not Report Paper Out: If this is selected, the printer driver will not report that paper is out even when paper out is sensed. This functionality is provided to maximize paper use on the Series 90 printer that reports paper out as a function of the number of print lines since paper low is sensed. Use on printers other than the Series 90 is not recommended. You must have printer configuration “Paper Out Mode” set to “Disabled” to use this functionality.

Series 90 Printers Only:

Front Slip: Set to do front slip validation on Series 90 printers

Top Form Right: Set to do top form (right justified) validation on Series 90 printers

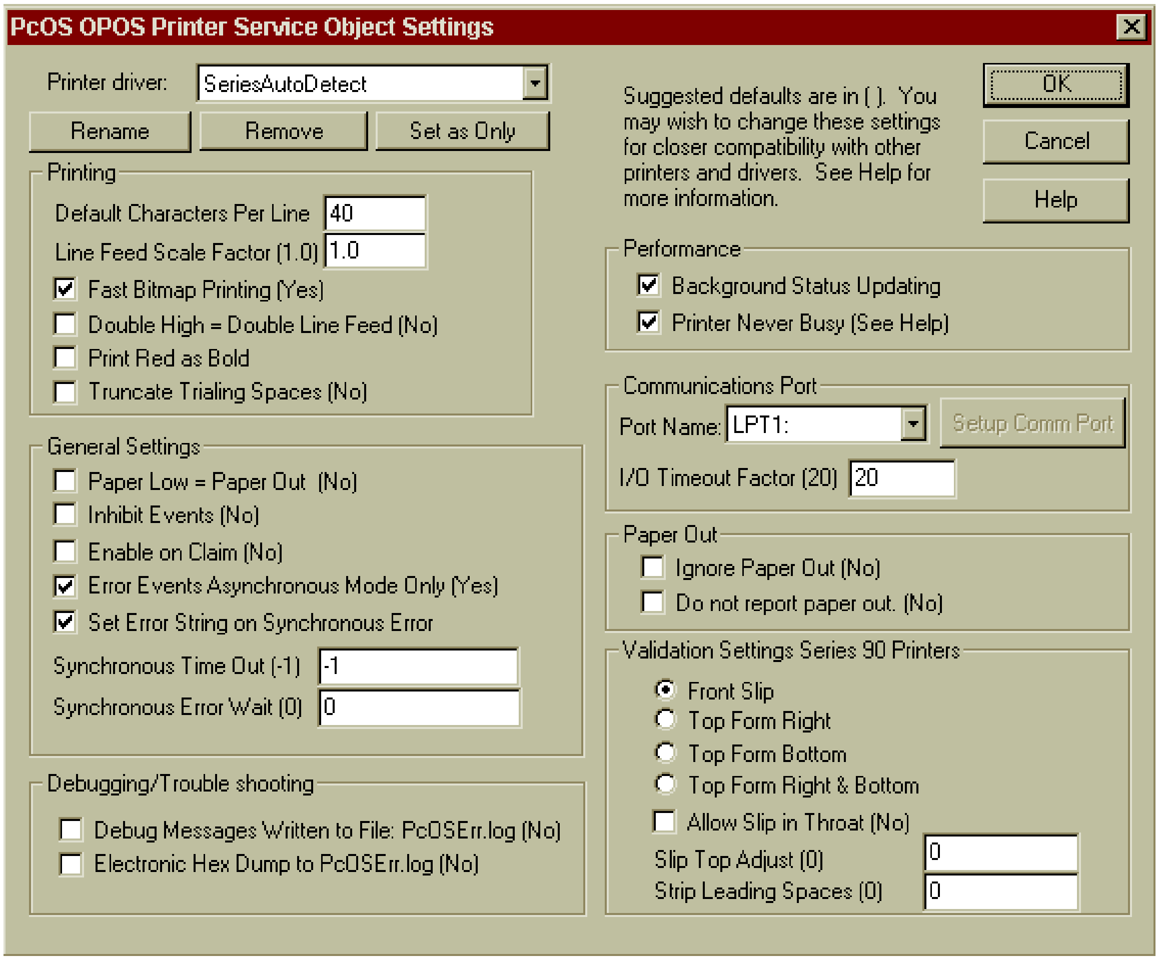
Top Form Bottom: Set to do top form (bottom justified) validation on Series 90 printers

Top Form Right & Bottom: Set to do top form (right & bottomed justified) validation on Series 90 printers

Allow Slip in Throat: Set to allow a form to be in the front slip "throat" of the printer when printing on a receipt.

Slip Top Adjust: This value controls the "Top of Slip" position when a slip is fed into a Series 90 printer. This setting can be used to control the position of the text on an inserted document. Units are in Dots/Pixels.

Strip Leading Spaces: This value controls how many spaces are removed from the beginning of a print line on a Slip or Form(non-rotated print). This capability is provided to allow replacement of printers that had a wider validation bed, where the validation text was space padded to position the text to the right. Units are in spaces.



Printer driver: SeriesAutoDetect

Suggested defaults are in ( ). You may wish to change these settings for closer compatibility with other printers and drivers. See Help for more information.

Printing

Default Characters Per Line: 40
Line Feed Scale Factor (1.0): 1.0
☑ Fast Bitmap Printing (Yes)
□ Double High = Double Line Feed (No)
□ Print Red as Bold
□ Truncate Trialing Spaces (No)

General Settings

□ Paper Low = Paper Out (No)
□ Inhibit Events (No)
□ Enable on Claim (No)
☑ Error Events Asynchronous Mode Only (Yes)
☑ Set Error String on Synchronous Error
Synchronous Time Out (-1): -1
Synchronous Error Wait (0): 0

Debugging/Trouble shooting

□ Debug Messages Written to File: PcOSErr.log (No)
□ Electronic Hex Dump to PcOSErr.log (No)

Performance

☑ Background Status Updating
☑ Printer Never Busy (See Help)

Communications Port

Port Name: LPT1:
I/O Timeout Factor (20): 20

Paper Out

□ Ignore Paper Out (No)
□ Do not report paper out. (No)

Validation Settings Series 90 Printers

● Front Slip
○ Top Form Right
○ Top Form Bottom
○ Top Form Right & Bottom
□ Allow Slip in Throat (No)
Slip Top Adjust (0): 0
Strip Leading Spaces (0): 0

Cash Drawer Dialog

Drawer 1 or 2: Select which drawer is the default cash drawer that is fired. This default setting can be overridden at run time via “DirectIO”. See: “PcOS OPOS Reference Manual” (PcOSOPOS.doc) for more information.

Drawer Open Status Immediately: If set, the driver will begin to report that the cash drawer is open as soon as the command is given to open the drawer. This option is provided to allow for applications that expect different event timing than the hardware generates.

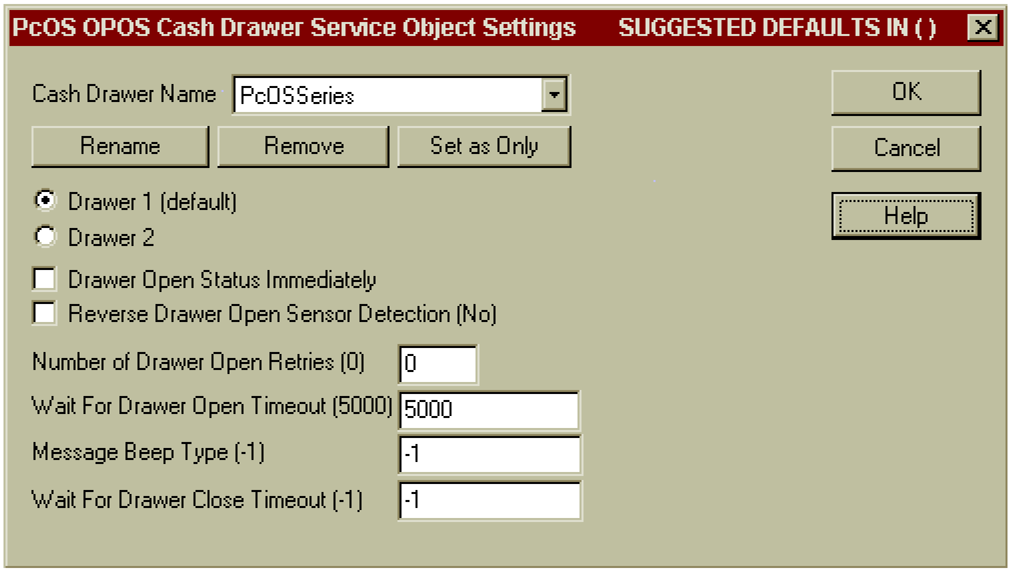
Reverse Drawer Open Sensor Detection: Some cash drawer sensors are wired opposite than expected (ie Normally open NO vs Normally closed NC). This setting allows for a cash drawer whose sensor is connected differently than expected.

Number of Drawer Open Retries: If non-zero and the cash drawer fails to open, the cash drawer driver will try to re-fire the cash drawer at approx. 2 seconds intervals the number of times specified.

Wait For Drawer Open Timeout: This is the time that the OpenDrawer() function will wait for the cash drawer to open. If the drawer does not open within this time, the function will return an OPOS_E_TIMEOUT. If set to -1 (0xffffffff), OpenDrawer() method will never time out.

Message Beep Type: Normally set to -1. When set to -1 will use PCs speaker as a beeper. This value is passed to the windows API MessageBeep() function.

Drawer Wait Timeout: Time in milliseconds that WaitForDrawerClose() method will wait before returning a OPOS_E_TIMEOUT. If set to -1 (0xffffffff), WaitForDrawerClose() method will never time out.



PcOS OPOS Cash Drawer Service Object Settings SUGGESTED DEFAULTS IN ( )

Cash Drawer Name: PcOSSeries

● Drawer 1 (default)
○ Drawer 2

□ Drawer Open Status Immediately
□ Reverse Drawer Open Sensor Detection (No)

Number of Drawer Open Retries (0): 0
Wait For Drawer Open Timeout (5000): 5000
Message Beep Type (-1): -1
Wait For Drawer Close Timeout (-1): -1

Printing via an OPOS driver

There are 3 Modes of printing.

1) Synchronous mode, printing is done before the print function returns, banking.
2) Asynchronous mode, print function returns as soon as it can, printing may or may not be done yet.
3) Transaction mode buffers all print data and the print data is sent all at once.

See the documentation that comes with the Ithaca OPOS drivers for detailed information on all the function calls that can be made via the OPOS drivers.

OPOS drivers are first “Opened”, “Claimed” and “Enabled” before they can be used. This is typically done when the application starts (Form Load in VB).

```
CallResult = Printer1.Open("SeriesAutoDetect")
Printer1.Claim (6000)
Printer1.DeviceEnabled = True
```

Then the device is used (i.e. printing is done)

```
Dim Data As String

‘SET NUMBER OF CHARACTERS PER LINE
Printer1.RecLineChars = 33

Data = "*********************************" + Chr(13) + Chr(10)
CallResult = Printer1.PrintNormal(PTR_S_RECEIPT, Data)
Data = "*                               *" + Chr(13) + Chr(10)
CallResult = Printer1.PrintNormal(PTR_S_RECEIPT, Data)
Data = "*         Fast Mart             *" + Chr(13) + Chr(10)
CallResult = Printer1.PrintNormal(PTR_S_RECEIPT, Data)
Data = "*                               *" + Chr(13) + Chr(10)
CallResult = Printer1.PrintNormal(PTR_S_RECEIPT, Data)
Data = "*********************************" + Chr(13) + Chr(10)
CallResult = Printer1.PrintNormal(PTR_S_RECEIPT, Data)
Data = Chr(13) + Chr(10) + Chr(10) + Chr(10) + Chr(10)
CallResult = Printer1.PrintNormal(PTR_S_RECEIPT, Data)

Data = Chr(27) + "|rA" + "Bread           1.49" + Chr(13) + Chr(10)
CallResult = Printer1.PrintNormal(PTR_S_RECEIPT, Data)
Data = Chr(27) + "|rA" + "Milk            2.19" + Chr(13) + Chr(10)
CallResult = Printer1.PrintNormal(PTR_S_RECEIPT, Data)
Data = Chr(27) + "|rA" + "Rice             .99" + Chr(13) + Chr(10)
CallResult = Printer1.PrintNormal(PTR_S_RECEIPT, Data)
Data = Chr(27) + "|rA" + "Soup            2.49" + Chr(13) + Chr(10)
CallResult = Printer1.PrintNormal(PTR_S_RECEIPT, Data)
Data = Chr(27) + "|rA" + "Nuts            3.49" + Chr(13) + Chr(10)
CallResult = Printer1.PrintNormal(PTR_S_RECEIPT, Data)
```

```
Data = Chr(27) + "|rA" + Chr(13) + Chr(10) + "----------------------------" + Chr(13)
+ Chr(10)
CallResult = Printer1.PrintNormal(PTR_S_RECEIPT, Data)
Data = Chr(27) + "|rA" + "TOTAL            10.65" + Chr(13) + Chr(10)
CallResult = Printer1.PrintNormal(PTR_S_RECEIPT, Data)

Data = Chr(13) + Chr(10) + Chr(10) + Chr(10) + Chr(10)
CallResult = Printer1.PrintNormal(PTR_S_RECEIPT, Data)

Data = "Thank you for Shopping with us" + Chr(13) + Chr(10)
CallResult = Printer1.PrintNormal(PTR_S_RECEIPT, Data)
Data = "Come again soon" + Chr(13) + Chr(10)
CallResult = Printer1.PrintNormal(PTR_S_RECEIPT, Data)


Data = Chr(27) + "|90fP"
CallResult = Printer1.PrintNormal(PTR_S_RECEIPT, Data)
```

When the device is no longer needed it is "Disabled", "Released" and "Closed". This is typically done when the application is being shutdown. (Form Unload in VB)**.**

```
Printer1.DeviceEnabled = False
Printer1.Release
Printer1.Close
```

Setting Fonts, Justification (left, right, center), Bold, Double wide/high, ... (from vbtest)

```
'Font Typeface #1
data = Chr(27) + "|1fT" + "Font Typeface #1" + Chr(10)
CallResult = Printer1.PrintNormal(PTR_S_RECEIPT, data)
'Font Typeface #2
data = Chr(27) + "|2fT" + "Font Typeface #2" + Chr(10)
CallResult = Printer1.PrintNormal(PTR_S_RECEIPT, data)
'Font Typeface #3
data = Chr(27) + "|3fT" + "Font Typeface #3" + Chr(10)
CallResult = Printer1.PrintNormal(PTR_S_RECEIPT, data)
'Font Typeface #0
data = Chr(27) + "|0fT" + "Font Typeface #0 (default)" + Chr(10)
CallResult = Printer1.PrintNormal(PTR_S_RECEIPT, data)


'Bold
data = "This is " + Chr(27) + "|bC" + "Bold" + Chr(10)
CallResult = Printer1.PrintNormal(PTR_S_RECEIPT, data)
'Underline
data = "This is " + Chr(27) + "|uC" + "Underline" + Chr(10)
CallResult = Printer1.PrintNormal(PTR_S_RECEIPT, data)
'Italic
data = "This is " + Chr(27) + "|iC" + "Italic" + Chr(10)
CallResult = Printer1.PrintNormal(PTR_S_RECEIPT, data)
```

```
'2xWide
data = "This is " + Chr(27) + "|2C" + "2xWide" + Chr(10)
CallResult = Printer1.PrintNormal(PTR_S_RECEIPT, data)
'2xHigh
data = "This is " + Chr(27) + "|3C" + "2xHigh" + Chr(10)
CallResult = Printer1.PrintNormal(PTR_S_RECEIPT, data)
'2xWide 2xHigh
data = "This is " + Chr(27) + "|4C" + "2xWH" + Chr(10)
CallResult = Printer1.PrintNormal(PTR_S_RECEIPT, data)
'2x Normal
data = "This is " + Chr(27) + "|4C" + "2x" + Chr(27) + "|1C" + " to Normal" + Chr(10)
+ Chr(10)
CallResult = Printer1.PrintNormal(PTR_S_RECEIPT, data)


'Left
data = "This is Left" + Chr(10)
CallResult = Printer1.PrintNormal(PTR_S_RECEIPT, data)
data = "This is More Left Text" + Chr(10)
CallResult = Printer1.PrintNormal(PTR_S_RECEIPT, data)
data = "This is Less" + Chr(10) + Chr(10)
CallResult = Printer1.PrintNormal(PTR_S_RECEIPT, data)

'Center
data = Chr(27) + "|cA" + "This is Centered" + Chr(10)
CallResult = Printer1.PrintNormal(PTR_S_RECEIPT, data)
data = Chr(27) + "|cA" + "This is More Centered Text" + Chr(10)
CallResult = Printer1.PrintNormal(PTR_S_RECEIPT, data)
data = Chr(27) + "|cA" + "This is Less" + Chr(10) + Chr(10)
CallResult = Printer1.PrintNormal(PTR_S_RECEIPT, data)


'Right Justified
data = Chr(27) + "|rA" + "This is Right Justified" + Chr(10)
CallResult = Printer1.PrintNormal(PTR_S_RECEIPT, data)
data = Chr(27) + "|rA" + "This is More Right Justified Text" + Chr(10)
CallResult = Printer1.PrintNormal(PTR_S_RECEIPT, data)
data = Chr(27) + "|rA" + "This is Less" + Chr(10) + Chr(10)
CallResult = Printer1.PrintNormal(PTR_S_RECEIPT, data)

'Paper feed (RecLinesToPaperCut) and cut
data = "Feed(RecLinesToPaperCut) cut" + Chr(10)
CallResult = Printer1.PrintNormal(PTR_S_RECEIPT, data)

data = Chr(27) + "|90fP"
CallResult = Printer1.PrintNormal(PTR_S_RECEIPT, data)
```

Opening a Cash drawer via an OPOS driver

```
Dim Count As Integer


CallResult = CashDrawer1.OpenDrawer()   'times out as per setting in control panel

'THE CODE BELOW THIS LINE IS OPTIONAL, but a good idea
Count = 5
If CallResult <> OPOS_SUCCESS Then
    While Not CashDrawer1.DrawerOpened And Count > 0
        Sleep (1000)
        Count = Count - 1
    Wend
End If

If Not CashDrawer1.DrawerOpened Then
    CallResult = CashDrawer1.OpenDrawer()
End If

If Not CashDrawer1.DrawerOpened Then
    MsgBox "Attempt to open the cash drawer failed."
End If

'THE CODE ABOVE THIS LINE IS OPTIONAL, but a good idea

If CheckWait.Value = 1 Then
CallResult = CashDrawer1.WaitForDrawerClose(3000, 0, 250, 250)
End If
```

Note: You must still Open, Claim and Enable the cash drawer driver, and Disable, Release and Close it.

QUIZ (OPOS Drivers)

What 3 types of OPOS drivers does Transact provide to our customers?

a) Big, Medium, and small.
b) Printer, Cash drawer, and MICR.
c) Larry, Curly, and Moe.

What do you have to do before you can print via an OPOS driver?

a) Install the drivers.
b) Configure the drivers.
c) Open, Claim and Enable the driver.
d) All of the above.

How do you set the CPI (character per inch) printed via a printer driver?

a) Send a printer escape code.
b) You don't
c) Via `Printer1.RecLineChars = ??`
d) b) and c)

How do I get the cash drawer to open?

a) Send an escape code via the printer driver.
b) Call `CashDrawer1.OpenDrawer()`
c) Use a big screwdriver and a hammer.

Why doesn't OPOS support a CPI (characters per inch) command?

a) They forgot to put it in.
b) The real issue is a consistent number of characters per line from printer to printer.
c) None of the above.

Section 4 (POSPrinter OCX, ActiveX)

Installing Ithaca POSPrinter OCX.

To start the install process, run the SetupPOS.exe file. Follow the on screen instructions. There is no configuration of the POSPrinter OCX.

Printing via POSPrinter OCX

See the documentation that comes with the POSPrinter OCX for detailed information on all the function calls that can be made with the POSPrinter OCX.

```
Dim StrData As String

Result = POSPrinter1.SetHandshake(2)
'Result = POSPrinter1.OpenPort("LPT1:", "")
Result = POSPrinter1.OpenPort("COM2:", "baud=9600 parity=N data=8 stop=1")
If Result <> 1 Then
MsgBox "OpenPort failed, make sure no other devices are using this port."
End If

StrData = "Hello World" + Chr(13) + Chr(10)

Result = POSPrinter1.SendString(StrData, Len(StrData))
If Result <> 1 Then
MsgBox "SendString failed, Ensure that the printer is connected to the port."
End If

Result = POSPrinter1.SendString(Chr(5) + Chr(21), 2)
StrData = POSPrinter1.ReadString(100)

Result = POSPrinter1.ClosePort()
```

QUIZ (POSPrinter OCX, ActiveX)

What are some disadvantages of using the POSPrinter OCX?

a) You have to understand the printer's command control codes.
b) You have to check to see if the printer got all the data you tried to send it.
c) You have to detect and recover from an error condition.
d) All of the above.
e) Both a) and c)
f) Both a) and b)
g) Both b) and c)

What are some advantages to using the POSPrinter OCX?

a) You can control exactly what gets sent to the printer.
b) You can get status back from the printer and interpret it yourself.
c) You can handle special cases that a driver may not address.
d) All of the above.
e) Both a) and c)
f) Both a) and b)
g) Both b) and c)

How do I make the printer print bold text using the POSPrinter OCX?

a) Send the bold escape command code.
b) Use the bold function command.
c) Push the "bold" button on the printer.
d) Any of the above.
e) None of the above.