



Zebra[®] Kiosk Driver

Reference Guide



© 2009 ZIH Corp. The copyrights in this manual and the software and/or firmware in the printer described therein are owned by ZIH Corp. Unauthorized reproduction of this manual or the software and/or firmware in the printer may result in imprisonment of up to one year and fines of up to \$10,000 (17 U.S.C.506). Copyright violators may be subject to civil liability.

This product may contain ZPL[®], ZPL II[®], and ZebraLink[™] programs; Element Energy Equalizer[®] Circuit; E³[®]; and Monotype Imaging fonts. Software © ZIH Corp. All rights reserved worldwide.

ZebraLink and all product names and numbers are trademarks, and Zebra, the Zebra logo, ZPL, ZPL II, Element Energy Equalizer Circuit, and E³ Circuit are registered trademarks of ZIH Corp. All rights reserved worldwide.

All other brand names, product names, or trademarks belong to their respective holders. For additional trademark information, please see “Trademarks” on the product CD.

Proprietary Statement This manual contains proprietary information of Zebra Technologies Corporation and its subsidiaries (“Zebra Technologies”). It is intended solely for the information and use of parties operating and maintaining the equipment described herein. Such proprietary information may not be used, reproduced, or disclosed to any other parties for any other purpose without the express, written permission of Zebra Technologies Corporation.

Product Improvements Continuous improvement of products is a policy of Zebra Technologies Corporation. All specifications and designs are subject to change without notice.

Liability Disclaimer Zebra Technologies Corporation takes steps to ensure that its published Engineering specifications and manuals are correct; however, errors do occur. Zebra Technologies Corporation reserves the right to correct any such errors and disclaims liability resulting therefrom.

Limitation of Liability In no event shall Zebra Technologies Corporation or anyone else involved in the creation, production, or delivery of the accompanying product (including hardware and software) be liable for any damages whatsoever (including, without limitation, consequential damages including loss of business profits, business interruption, or loss of business information) arising out of the use of, the results of use of, or inability to use such product, even if Zebra Technologies Corporation has been advised of the possibility of such damages. Some jurisdictions do not allow the exclusion or limitation of incidental or consequential damages, so the above limitation or exclusion may not apply to you.

Contents



1 • Introduction	5
Contacts	6
3 • Printing Preferences	7
Paper/Output	9
Paper Size	9
Copy Count	10
Document Options	11
Printer Features	12
4 • Device Settings	21
5 • Ports	23
Enable Bi-Directional Support	24
Configure Port	25
6 • Additional Properties Tabs	27
Printer Keys	28
Monitor Keys	29
7 • Language Monitor and Advanced Driver Properties Restrictions	31
Using Cut Per Document With Page Hold Enabled	31
Retract Behavior Set To “Eject” With Page Hold Enabled	31
Restarting A Print Job During Cut/Eject	31
A Presenter Jam Doesn’t Clear All Print Jobs	31
Offline/Online Status Is Not Automatically Updated	32

8 • Bi-Directional Communication	33
The Language Monitor	33
Zebra Functions for Communication with the Printer	34
Windows APIs for Communication with the Printer	36
GetPrinter	39
ATL Object for Communication with the Printer	44
ATL Object Function and Property Definition	44
Event Notification	46
Registry Entries	47
In the Printer Key	48
A • Status Messages	49
B • External Error Status Messages	51
C • Programming Examples	53
Background	53
Monitoring while printing	54
Monitoring while idle	54
Implementation	55
Open the Printer	55
Open the Event Handles	55
Start Monitoring	56
Fill Event Arrays	56
Start the Waiting Loop	57
Index	61



Introduction

The TTP printer family consists of the following printers.



TTP 2000 series



TTP 2100 series



TTP 7000 series



TTP 8000 series

This document describes the drivers for Windows 2000 and XP, which are all based on Microsoft Unidriver with special OEM plugins.

Contacts

Technical Support via the Internet is available 24 hours per day, 365 days per year.

Web Site: www.zebra.com

E-mail Back Technical Library:

E-mail address: emb@zebra.com

Subject line: Emaillist

Self Service Knowledge Base: www.zebra.com/knowledgebase

Online Case Registration: www.zebra.com/techrequest

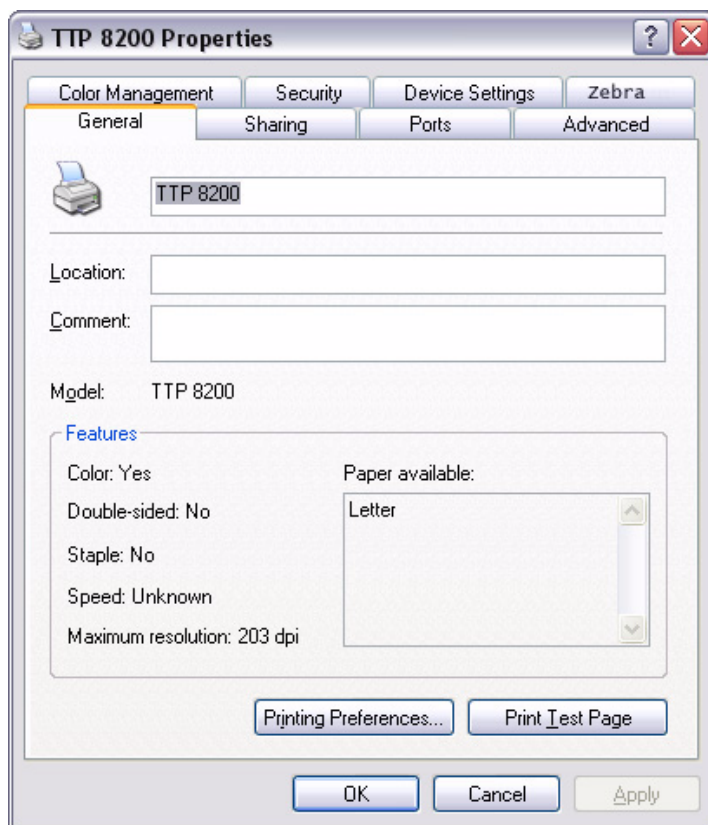
Which Department Do You Need?	The Americas	Europe, Middle East, and Africa	Asia Pacific and India
Regional Headquarters	Zebra Technologies Corporation 475 Half Day Road, Suite 500 Lincolnshire, IL 60069 USA T: +1 847 634 6700 Toll-free +1 866 230 9494 F: +1 847 913 8766	Zebra Technologies Europe Limited Dukes Meadow Millboard Road Bourne End Buckinghamshire, SL8 5XF United Kingdom T: +44 (0) 1628 556000 F: +44 (0) 1628 556001	Zebra Technologies Asia Pacific Pte. Ltd. 120 Robinson Road #06-01 Parakou Building Singapore 068913 T: +65 6858 0722 F: +65 6885 0838
Technical Support For questions on the operation of Zebra equipment and software, please call your distributor. For additional assistance, contact us. <i>Please have your model and serial numbers available.</i>	T: +1 877 ASK ZEBRA (275 9327) F: +1 847 913 2578 Hardware: ts1@zebra.com Software: ts3@zebra.com <i>Kiosk printers:</i> T: +1 866 322 5202 E: kiosksupport@zebra.com	T: +44 (0) 1628 556039 F: +44 (0) 1628 556003 E: Tseurope@zebra.com	T: +65 6858 0722 F: +65 6885 0838 E: China: tschina@zebra.com All other areas: tsasiapacific@zebra.com
Repair Service Department For back-to-base service and repair.	T: +1 877 ASK ZEBRA (275 9327) F: +1 847 821 1797 E: repair@zebra.com To request a repair in the U.S., go to www.zebra.com/repair .	T: +44 (0) 1772 693069 F: +44 (0) 1772 693046 New requests: ukrma@zebra.com Status updates: repairupdate@zebra.com	T: +65 6858 0722 F: +65 6885 0838 E: China: tschina@zebra.com All other areas: tsasiapacific@zebra.com
Technical Training Department For Zebra product training courses.	T: +1 847 793 6868 T: +1 847 793 6864 F: +1 847 913 2578 E: tamerica@zebra.com	T: +44 (0) 1628 556000 F: +44 (0) 1628 556001 E: Eurtraining@zebra.com	T: +65 6858 0722 F: +65 6885 0838 E: China: tschina@zebra.com All other areas: tsasiapacific@zebra.com
Inquiry Department For product literature and distributor and dealer information.	T: +1 877 ASK ZEBRA (275 9327) E: inquiry4@zebra.com	T: +44 (0) 1628 556037 F: +44 (0) 1628 556005 E: mseurope@zebra.com	E: China: GCmarketing@zebra.com All other areas: APACChannelmarketing@zebra.com
Customer Service Department (US) Internal Sales Department (UK) For printers, parts, media, and ribbon, please call your distributor or contact us.	T: +1 877 ASK ZEBRA (275 9327) E: clientcare@zebra.com	T: +44 (0) 1628 556032 F: +44 (0) 1628 556001 E: cseurope@zebra.com	T: +65 6858 0722 F: +65 6885 0836 E: China: order-csr@zebra.com All other areas: csasiapacific@zebra.com

Key: T: Telephone
F: Facsimile
E: E-mail

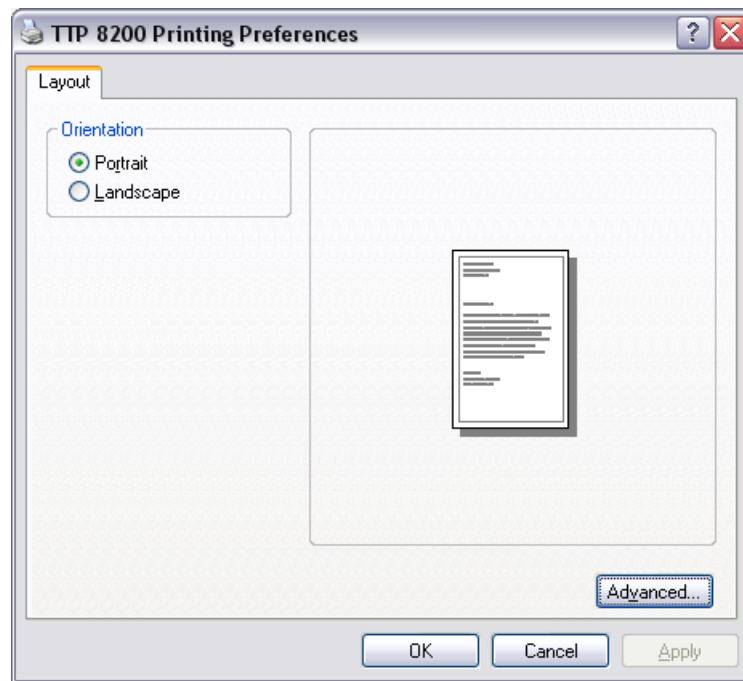


Printing Preferences

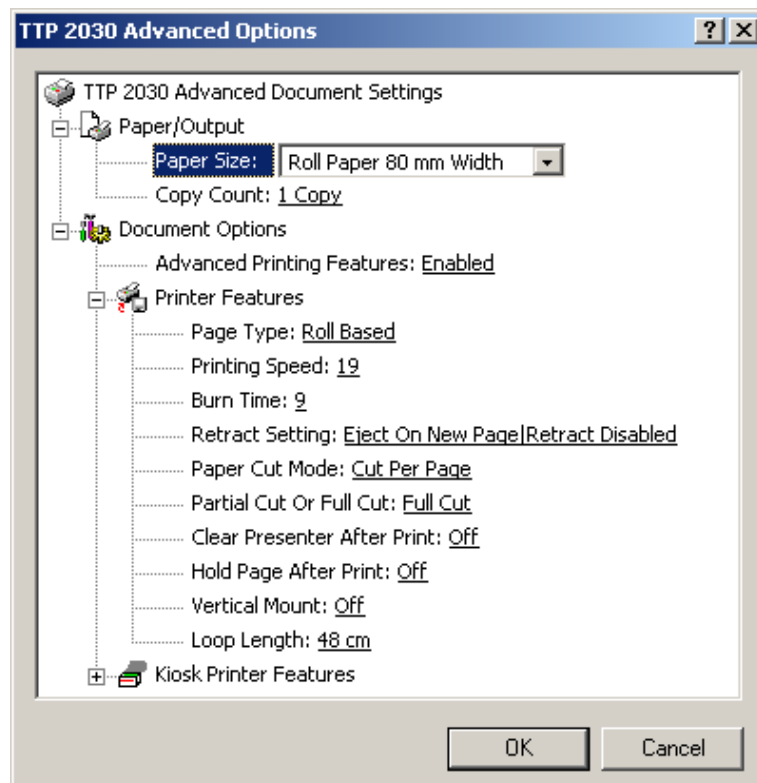
You can reach the Printing Preferences either through the button in the Printer Properties or directly through the right-click printer menu in the Printers Control Panel or the Printers and Faxes window (Windows XP).



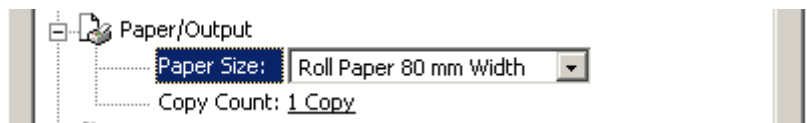
In Printing Preferences you can only select Portrait or Landscape orientation; all other settings are under the Advanced button.



If you select the Advanced button, the Advanced Options allow you to modify printing behavior and quality.

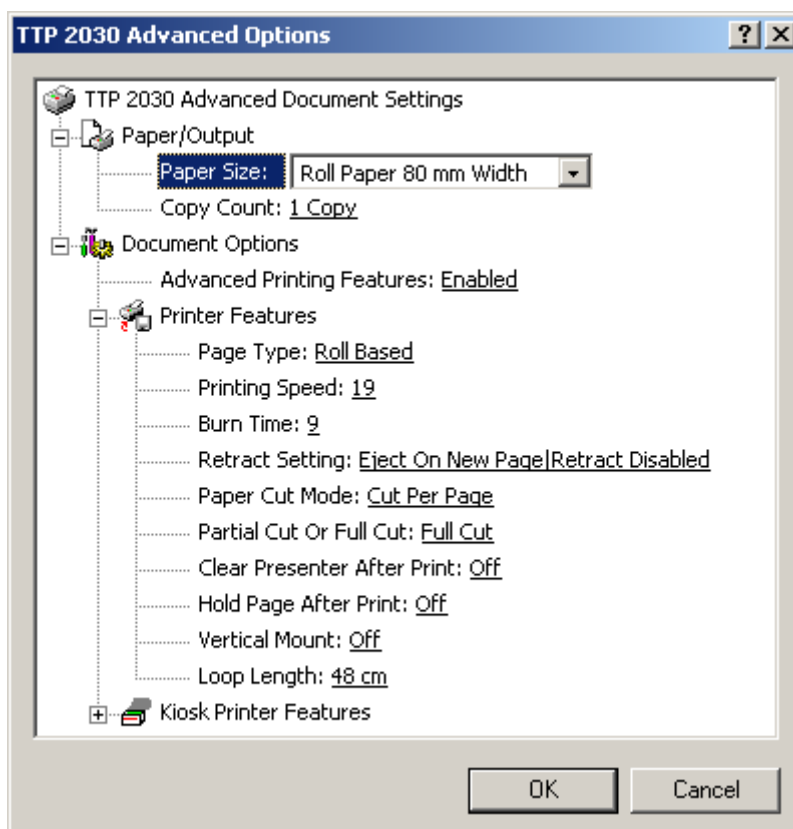


Paper/Output

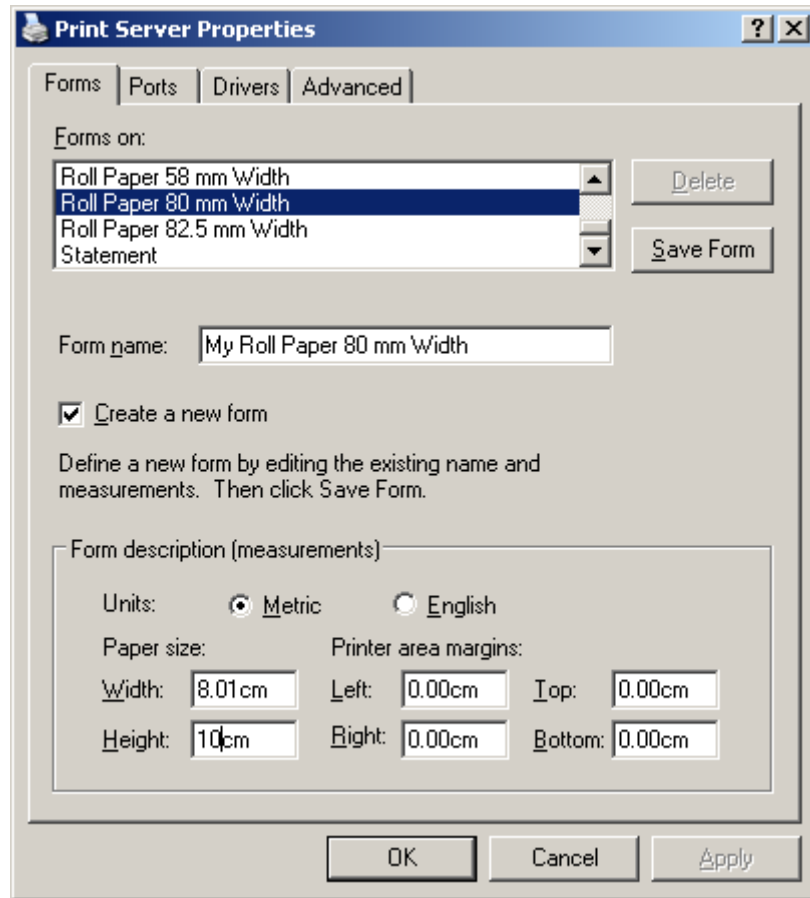


Paper Size

The paper size allows you to adjust the printer default paper size to any one of the forms available by default.



In addition to the currently available forms, you can create new ones in the Server Properties dialog. To access this dialog, right click on a blank area in the Printers Control Panel and select Server Properties from the menu that appears.



Select the form closest to what you want as your starting point and modify it with your customizations. Give the form a new name (*never* overwrite the original form) and click Save Form.

You may modify the height as well as the left and right margins.

Copy Count

Copy count prints multiple copies of each page sent. The printer can't store and reproduce the page sent to it, so this is handled by the host PC.

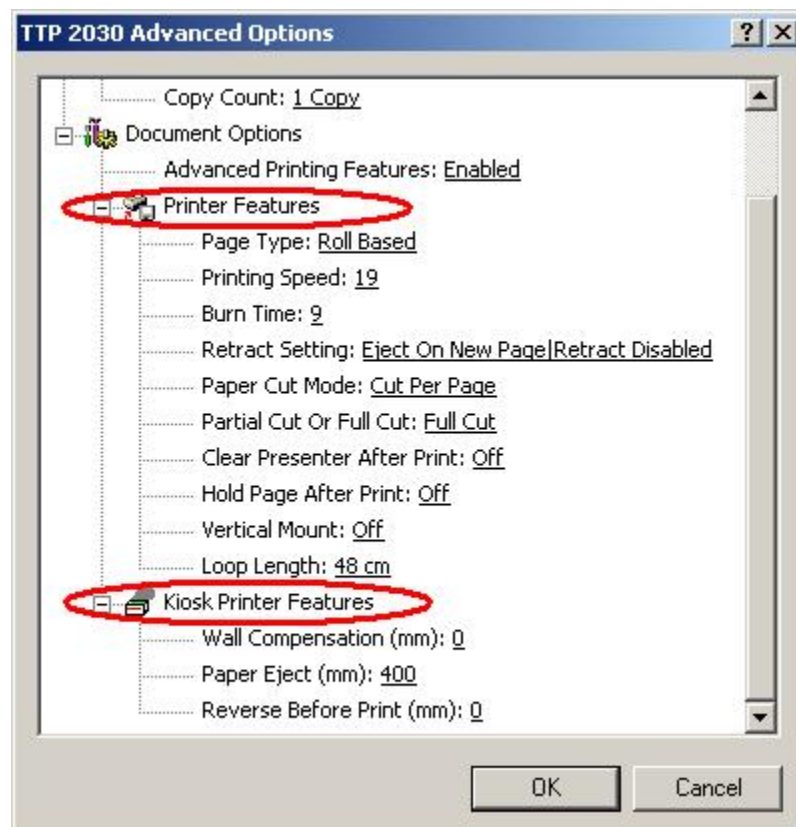
Document Options

Some of the document options override parameter settings stored in the printer. When the printer is turned off and back on, the overridden parameters return to their default value. This is normally not a problem because the selections you make in Windows are sent to the printer with each page printed. However, it is an advantage to have the same parameters both stored in the printer and set up in Windows.

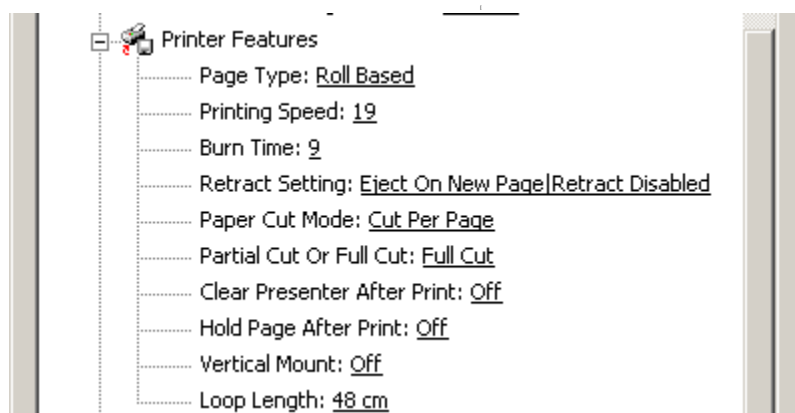


Note • The Windows settings are not used during operations such as paper loading, so if you are using paper with black registration marks, it is recommended to have your black mark configuration set in the printer as well as in the Document Options. This way, auto loading of paper will work correctly even if the printer has been turned off.

The Document Options dialog includes two printer-specific parts: the Printer Features and the Kiosk Printer Features.



Printer Features



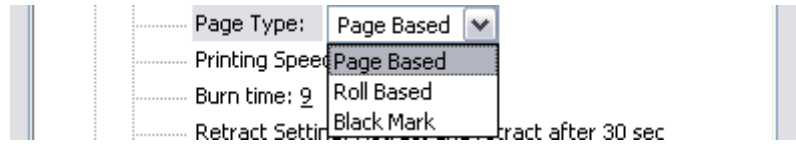
The following table shows the driver settings and their corresponding printer parameters:

Table 1 • Printer Parameters

Driver Setting	Printer Parameter	Parameter Number
Page Type	Document Mode	36
Printing Speed	Max print speed	8
Burn time	Burn time	7
Retract Setting	Waste basket mode	45
Paper Cut Mode	—	—
Partial Cut Or Full Cut	Partial Cut Length	60
Clear Presenter After Print	—	—
Hold Page After Print	—	—
Vertical Mount	System	57
Loop Length	Presenter loop length	9

Page Type

The Page Type selection will control the cutting position.



- **Page Based**

The printer will always feed as much paper as it needs for the set default Paper Size and then cut the paper at the end of the page.



Example • If you have four inches of text and you have “Letter” selected as your paper size, you will get a Letter-sized (11in) page.

- **Roll Based**

The printer will cut the page after the last line of text plus any additional paper advancement set in the Advanced Feature section.



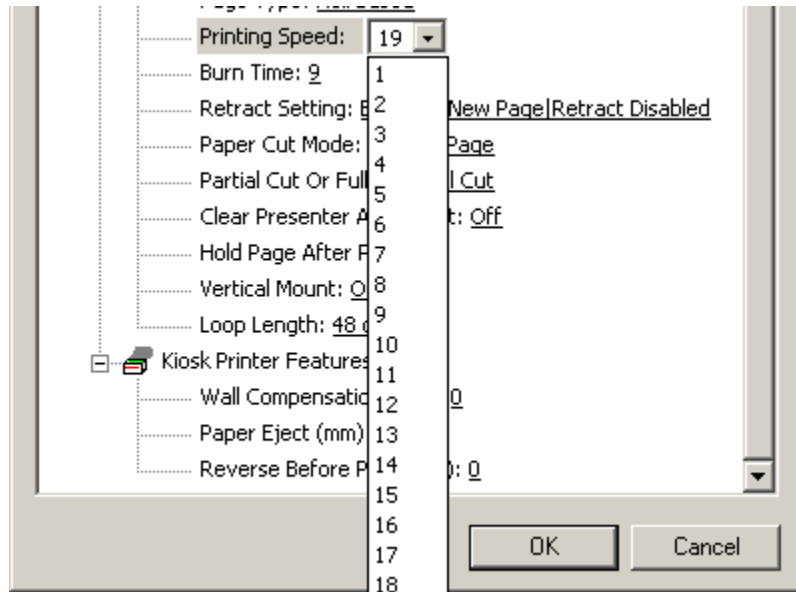
Example • If you have four inches of text, the printer will end the page after this text.

- **Black Mark**

The cut will be controlled through the Black Mark settings and the Black Marks on the underside of the paper. (Also known as registration or sense marks).

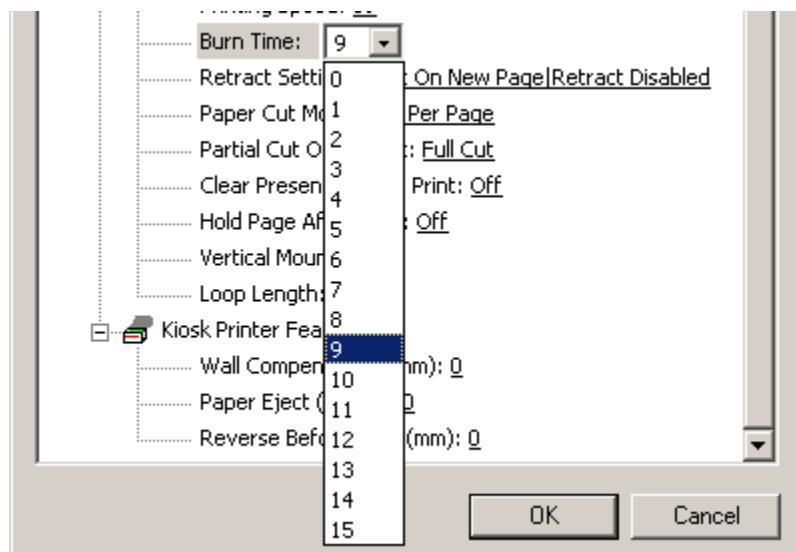
Printing Speed

The printing speed setting will change the actual print speed according to the value selected and as referenced in the Installation / Technical Manual for your printer. You rarely need to change this setting, as the default is the maximum speed.



Burn Time (Paper Sensitivity)

Burn time will change the burn time setting in the printer according to the value selected and as referenced in the Installation /Technical Manual for your printer. A higher value gives more dense (darker) print, but may also slow down printing and make barcodes unreadable.



Retract Setting

The Retract Setting controls the Waste basket mode in the printer and sets the value of parameter 45 according to the Technical Manual. This setting consists of two parts: what occurs when a new page is sent to the printer, and what occurs after a given timeout. For example, the selection “Retract |Retract 30 sec” will retract the page currently in the presenter when a new page arrives. If no new page arrives, any page remaining in the presenter after 30 seconds will be retracted. The options “Eject” and “Retract” do not have any timeout action; the page will be ejected or retracted when a new page arrives to be printed.

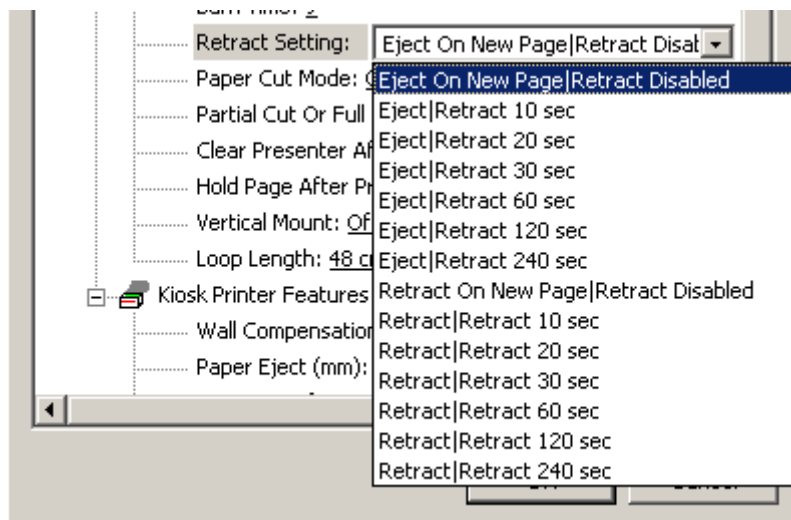
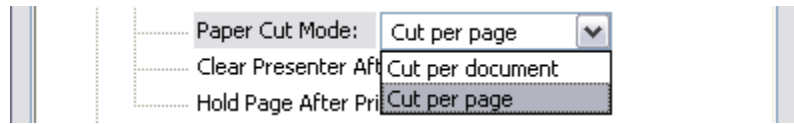


Table 2 • Retract and Retain Settings Overview

What to do with uncollected printouts remaining in presenter when a new page is to be printed:	When the customer does not take the printout:	After what time is this action performed:	Setting:
Eject	Do nothing	Disabled	Eject
Retract	Retract	10 s	Eject and retract after 10 sec
Retract	Retract	20 s	Eject and retract after 20 sec
Retract		Etc.	Etc.
Retract	Do nothing	Disabled	Retract
Retract	Retract	10 s	Retract and retract after 10 sec
Retract	Retract	20 s	Retract and retract after 20 sec
Retract		Etc.	Etc.
Hold until taken *	Retract	10 sec	Retract and retract after 10 sec, and Hold page after print = ON

* Holds the subsequent pages in a print job until the completed & presented page is taken by the customer. If a page is not taken before the timeout elapses, the entire print job is deleted from the Windows spooler. This function is not available on all printers.

Paper Cut Mode



Paper Cut Mode allows control of the way the printer operates after a Page End or Document End. “Cut per document” will feed as many pages uncut through the presenter as the document includes and cut at the end of the document.

➔ **Example** • If your print job consists of 3 pages, you will end up with one long uncut sheet which includes these three pages.



Note • If you intend to feed long documents: you can only loop a maximum of 60 cm (23 in.) inside the presenter. When this amount has built up, the printer will start feeding out the complete document while printing. The amount of paper looped is controlled by printer parameter 9, “presenter loop length”. Please see the technical manual for more information. Also, documents larger than 4 m (100 in.) will be cut after this length in order to assure proper printer functionality.

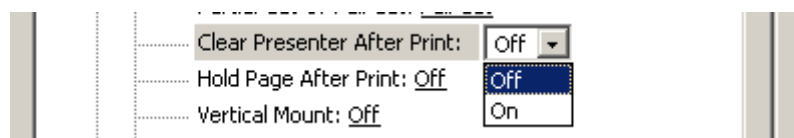
“Cut per page” will cut every page (as defined by the selected page size) of a printed document and present the existing page after the next page has started printing.

➔ **Example** • If your print job consists of 3 pages, you will end up with 3 individually cut sheets.



Note • If you print documents with more than one page you need to make sure that the paper can be collected outside the printer, since a completed page will be ejected from the printer when the next page prints. Ensure that the wastebasket setting is set to “Eject...” or else each page of your document will be retracted back into the enclosure when a subsequent page prints.

Clear Presenter After Print



When set to “On” this function enables a forced full eject after a page is printed and cut. If set to “Off” the page will only be partially ejected, and the retract settings are in effect.

Hold Page After Print

This function holds the page in the presenter until the customer takes it even if new pages are waiting to be printed. When the customer takes a page, the next page is printed, and so on.

If the customer does not take a page, the page remains in the presenter until the retract timer elapses, then the presenter is cleared according to the behavior specified in the Retract Setting and the rest of that print job is deleted from the Windows print spooler.



Note • The “Hold Page After Print” function relies on delicate interaction between the printer and Windows, so bi-directional communication must be on and working before you enable this function.

“Hold Page After Print” only works in “Cut Per Page” mode (because it needs individual pages to hold).

Vertical Mount

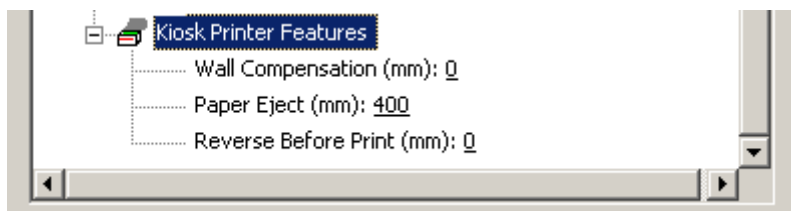
The TTP 20x0 printer supports vertical mounting of the printer. In this mode, the normal paper exit is used as a retract slot, and the normal retract slot becomes the paper exit. To activate this mode, change the Vertical Mount option to On.

Loop Length

This parameter controls the loop length in the printer and modifies the parameter 9 in the printer properties. Limits the maximum loop length. When the set length is reached, the printer ejects part of the printout and continues to print. You use this when you have very limited spaces for the loop inside the kiosk. Each step represents a 3.2 cm increment.

Kiosk Printer Features

There are three OEM kiosk features.

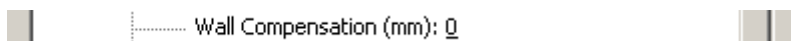


The following table shows the driver settings and their corresponding printer parameters or commands.

Table 3 • Printer Commands and Parameters

Driver Function	Printer Command
Wall Compensation	Parameter 47
Paper Eject	Advances paper using ESC FF after it has been cut. Corresponds to printer parameter 47.
Reverse Before Print	Use ESC j to reverse paper.

Wall Compensation



When the printout is printed and cut, the presenter ejects the amount defined by Paper Eject so that the customer can grab it. If the kiosk wall is thick, or if you just want a longer part of the printout to be visible, this parameter adds extra eject length.

Paper Eject



Paper eject will allow control of the amount of paper fed out of the printer after cutting. Adjust this setting to work with the thickness of your kiosk wall.



Note • Some printer families like the TTP 1000 and TTP 70x0 have a DC motor controlling the presenter and the value is approximate.

Reverse Before Print



The Reverse Before Print setting allows the user to control the Top Margin of the document. Since the printer has a physical distance between the print head and the cutter of either 14 mm (for TTP 1000 and TTP 7000), 9 mm (for TTP 2000 and TTP 2100) or 19 mm (for TTP 8000) you will always encounter a Top Margin of 14 mm, 9 mm, or 19 mm respectively. This can be reduced with this function.

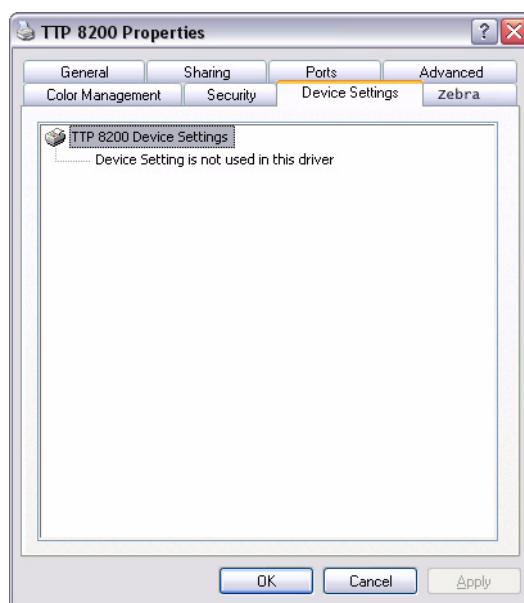
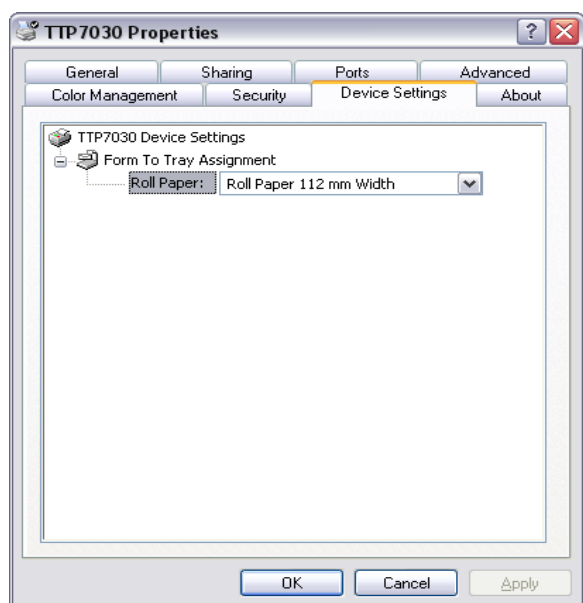


Note • To ensure printing safety and avoid paper jams, verify that the paper is not pulled back too far. A minimum Top Margin of 2 mm must be kept to avoid paper jams. Do not set this value to greater than 12 mm for TTP 1000 and TTP 7000, 7 mm for TTP 2000 and TTP 2100, or 17 mm for TTP 8000.



Notes • _____

Device Settings



Device settings are used for the TTP 1000, TTP 2000, TTP 2100, and TTP 7000 series printers only. Device settings are not used for the TTP 8000 series.



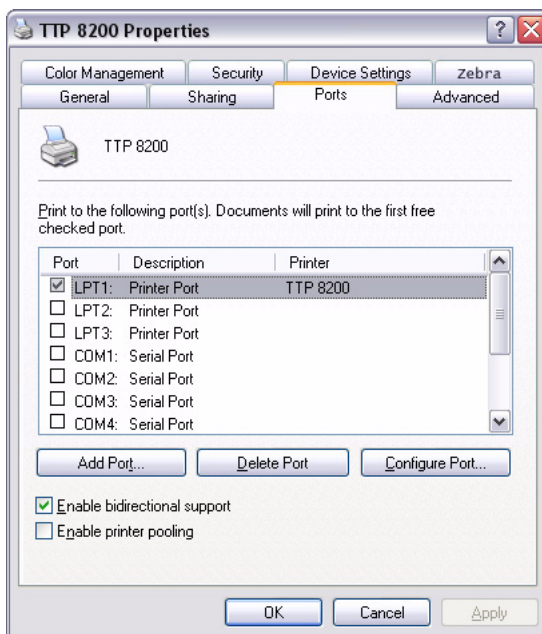
Notes •



The Ports option is used to setup the IO port used to print to the printer and also to configure the serial IO if used.

The following port types are usually available: LPT, COM, USB.

- LPT (parallel) port is commonly used and doesn't need any setup. The printers TTP 1020, TTP 2020, TTP 7020, and TTP 8200 can exchange status information via the parallel port by means of the Language Monitor interface.
- COM (RS232 serial) port is generally an option typically used in text applications where no Windows driver is used, due to its slow transfer speed. The Language Monitor functions are not fully supported when using the RS232 interface.
- USB is a fast serial IO. It can be used with TTP 1030, TTP 2030, TTP2130, TTP 7030, and TTP 8200. The USB port also allows exchanging status information by means of the Language Monitor interface.



Enable Bi-Directional Support

This selection enables use of the Language Monitor.

The Language Monitor is an additional DLL in the Windows driver chain that allows the printer to exchange status information with the PC. In order to use the Language Monitor it is necessary to program support for it within an application.



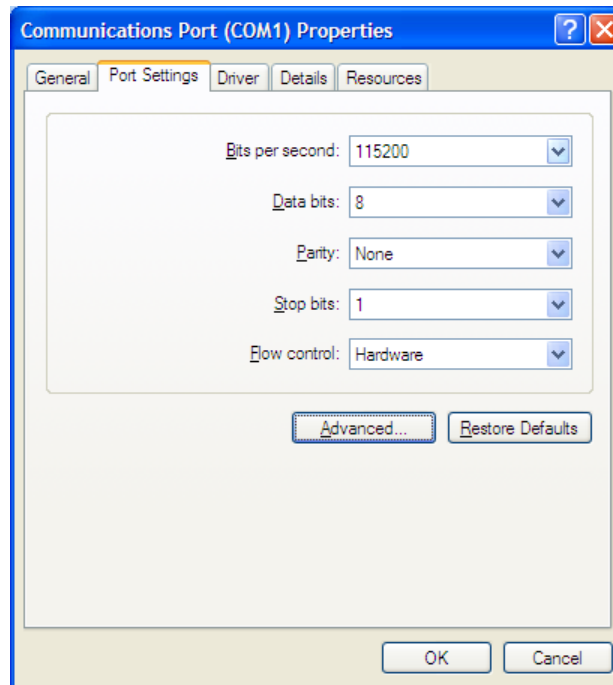
Note • The printer should be powered on before the PC to ensure the language and status monitor function correctly and report the correct status.



Note • For the printer to operate under all error situations, the printer parameter 5 “Disable parallel port signaling” must be set to 1 when using the parallel interface. The Technical Manual describes how to set parameters in the printer and utility programs for this are available from Zebra.

Configure Port

Port settings are available if you use the serial COM port or the parallel LPT port. This port is not commonly used with the Windows driver because of its limited transfer speed.



There is one option under Configure Port when using the parallel interface, Transmission Retry timeout. In the event that there is a problem in communicating with the parallel printer, Windows will suspend port activity for this duration. After this timeout has elapsed, it will retry communication.

- Bits per second
This setting must match the Baud rate setting in the printer parameters (1).
- Data bits
This setting must match the Data bits setting in the printer parameters (2).
- Parity
This setting must match the Parity setting in the printer parameters (3).
- Stop bits
The printer is fixed at 1 stop bit.
- Flow control
This setting has to match the Flow control setting in the printer parameters (4).

When using the Windows driver, you must always use Hardware flow control. Graphical page data may very well contain the XON (11h) or XOFF (13h) characters which will interrupt data transfer.



Note • The port settings have to match the settings in the printer based on parameters 1 through 4.

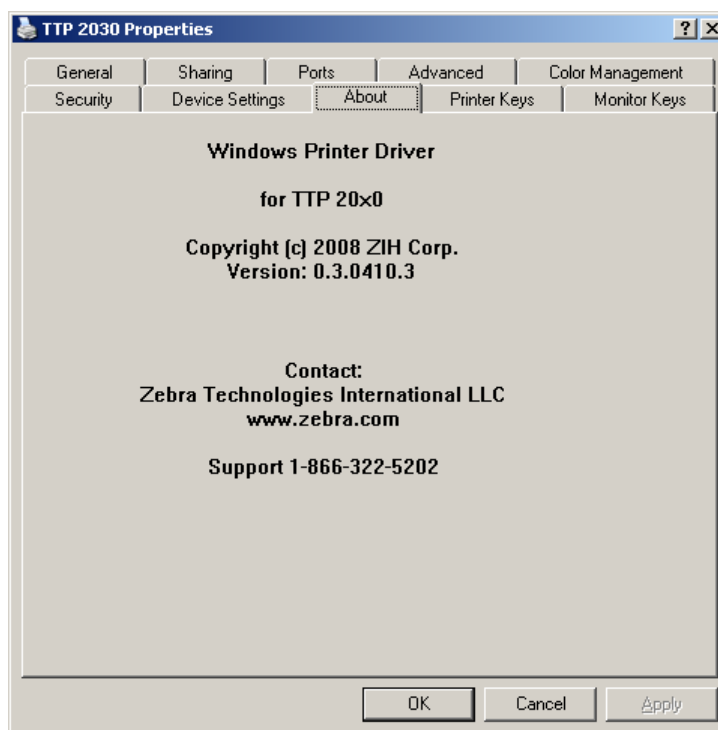


Notes • _____



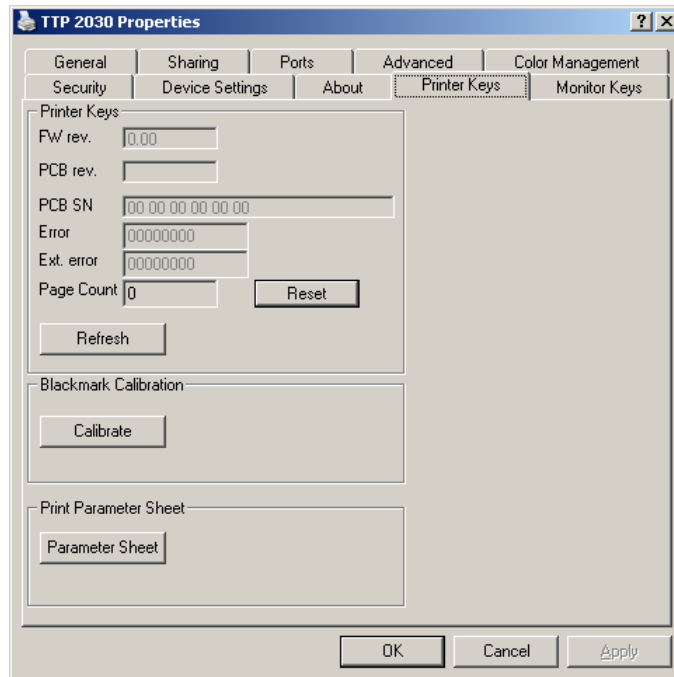
Additional Properties Tabs

Depending on your printer model and driver version, you may have two additional Properties Dialog Tabs, Printer Keys and Monitor Keys:



These two tabs provide an interface to the registry-based events, status, and settings used in the Zebra Windows Driver.

Printer Keys



Note • The printer should be powered on before the PC to ensure the language and status monitor function correctly and report the correct status.

- The Printer Keys tab displays printer information that is stored in the registry.
- “FW rev.” displays the firmware version loaded on the printer.



Note • The Language Monitor cannot always obtain this information, especially on parallel printers.

- “PCB rev.” displays the revision of the printer control board.
- “PCB SN” displays the serial number of the PCB.



Note • The PCB SN is NOT the serial number of the printer! The serial number of the printer is found on a printed label on the side of the printer and cannot be obtained through software.

- “Error” is the printer status code, as described in [Bi-Directional Communication on page 33](#).
- “Ext error” is the printer extended status code, as described in [Bi-Directional Communication on page 33](#).
- “Page Count” displays the number of pages that have been printed since the driver was installed or the counter was reset. To set the counter manually, type a new page count and click the Change button.

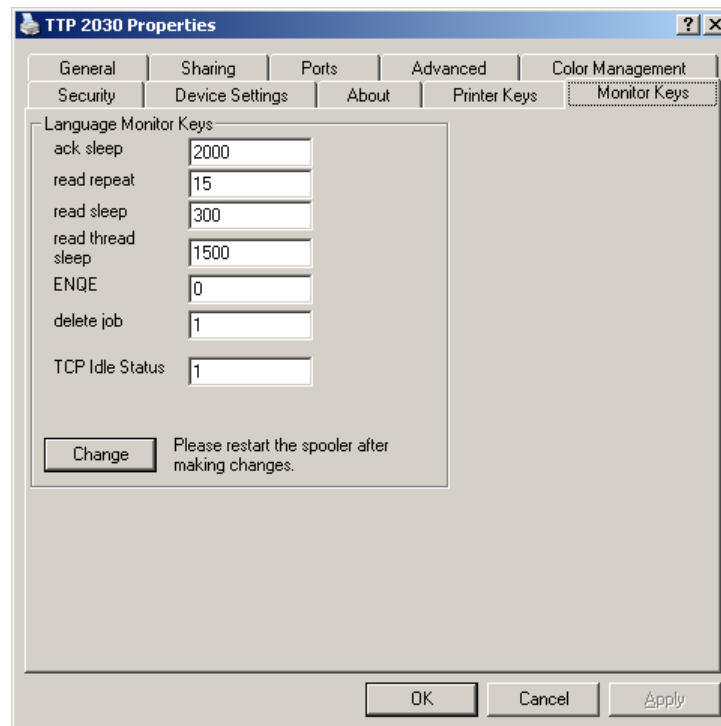


Note • This value is current as of the time the Printer Properties Dialog is opened. You can get an updated status by clicking the “Refresh” button.

To calibrate your printer for use with Black Mark media, refer to the calibration instructions in the Technical Manual for your printer.

If you wish to find out the current firmware loaded on the printer or any of the other internal parameters, you may print a self-test parameter sheet by clicking the “parameter sheet” button.

Monitor Keys



The Language Monitor keys tab allows changes to internal Language monitor.



Important • Do no alter these values unless instructed by Zebra Technical Support!

No further explanation will be given here.



Notes •



Language Monitor and Advanced Driver Properties Restrictions

Using Cut Per Document With Page Hold Enabled

When cut per document is on, there is no cut and eject between pages of a job. The page hold function waits for a page of a job to be taken before continuing. Since there was no cut/eject, the user can't take the page, and the Language Monitor (LM) waits indefinitely (freezes). A reboot and a printer power cycle are necessary. Even then, the job may not be able to be deleted easily.

Solution: Don't allow page hold function to be used with cut per document mode.

Retract Behavior Set To "Eject" With Page Hold Enabled

The LM will hold the job pending the retrieval of the page. If the page is not taken, the printer ejects the page after about 1 minute. The next page prints, but the LM hangs. A reboot and a printer power cycle are necessary, and even then, the job may not be able to be deleted easily.

Solution: The LM is not receiving the NAK 16 signal, since there was no official timeout and retract/eject. Do not use these settings together. Use one of the two-part settings, such as Eject and Retract after 60 seconds.

Restarting A Print Job During Cut/Eject

Restarting a print job while the job is in the cut/eject phase may lock the LM and require a reboot.

Solution: Under normal printing circumstances, this should never happen.

A Presenter Jam Doesn't Clear All Print Jobs

When a presenter jam occurs, the Language Monitor will not delete all print jobs in the spooler, only the current print job (the job that encountered the error). All pending print jobs will be paused. After the error is cleared the spooled jobs will continue printing.

Offline/Online Status Is Not Automatically Updated

When the printer is powered off and powered on again, the printer status may not update in the Windows Spooler. Also, if the printer is restarted with a pending error, or if the spooler is restarted with a pending error, the status may not automatically update.

Solution: When a printer goes offline, Windows deactivates the Language Monitor used for that printer. Language monitors are not restarted until a bi-directional operation comes through the spooler. This operation can be a print job sending an ENQ (clear presenter) command to the printer.



Bi-Directional Communication

The Language Monitor

The Windows operating system beginning with Windows NT4 had major restrictions with regard to bi-directional communication when using a Windows driver. In Windows 98 you could use a driver and still open the same port to write and read status information to it. After Windows NT4, however, you need a Language Monitor to do the same thing.

The Language Monitor is part of the Windows driver and sits between the Driver UI and the Port Monitor, which takes care of the direct communication with the selected port.

The Zebra Language Monitor has a Windows API interface through the function `GetPrinterData` and `GetPrinter`. It also offers event notification when status of the printer changes.

See the Appendix for a description and programming example.

Zebra Functions for Communication with the Printer

Windows Compatible Status with Auto Status

These statuses will also be stored in the printer ERROR key in the registry and can be extracted with GetPrinterData.

Table 4 • Windows Status

Windows Status	Compares to Zebra Status
PRINTER_STATUS_PAPER_JAM	Paper jam (ESC ENQ 1 = 1)
PRINTER_STATUS_ERROR	Temperature error (ESC ENQ 1 = 6)
PRINTER_STATUS_PAPER_PROBLEM	Paper feed problem (ESC ENQ 1 = 5)
PRINTER_STATUS_DOOR_OPEN	Print head lifted (ESC ENQ 1 = 4)
PRINTER_STATUS_PAPER_OUT	Out of paper (ESC ENQ 1 = 3)
PRINTER_STATUS_USER_INTERVENTION	Cutter not home (ESC ENQ 1 = 2)
PRINTER_STATUS_PAPER_NEAR_END	Paper near end (ESC ENQ 6)
PRINTER_STATUS_PAPER_WEEKEND	Weekend paper status (ESC ENQ 6)
PRINTER_STATUS_ERROR	Undefined error

Statuses Defined in winspool.h

Table 5 • Status Defined in Winspool

#define PRINTER_STATUS_ERROR	0x00000002
#define PRINTER_STATUS_PAPER_JAM	0x00000008
#define PRINTER_STATUS_PAPER_OUT	0x00000010
#define PRINTER_STATUS_PAPER_PROBLEM	0x00000040
#define PRINTER_STATUS_OFFLINE	0x00000080
#define PRINTER_STATUS_USER_INTERVENTION	0x00100000
#define PRINTER_STATUS_DOOR_OPEN	0x00400000

Statuses Defined by Zebra

Table 6 • Status Defined by Zebra

#define PRINTER_STATUS_PAPER_NEAR_END	0x02000000
#define PRINTER_STATUS_PAPER_WEEKEND	0x04000000
#define PRINTER_STATUS_PAPER_AT_PRESENTER	0x08000000
#define PRINTER_STATUS_EXTERNAL_ERROR	0x10000000



Note • The Zebra defined status results may be used by Microsoft Operating Systems with later Service releases. This may cause erratic status responses and will be addressed in a later version of this driver..

Windows Compatible Status with Ext Auto Status

These statuses will also be stored in the printer EXTERNALERROR key in the registry and can be extracted with GetPrinterData.

Below statuses are Extended Error statuses defined by Zebra. For the meanings of these NAK responses, please see the appropriate Technical Manual for your printer, under the ESC ENQ 1 section.

Table 7 • Extended Error Status

#define NAK6	0x00000001
#define NAK7	0x00000002
#define NAK12	0x00000004
#define NAK13	0x00000008
#define NAK14	0x00000010
#define NAK16	0x00000020
#define BUFFEROVERFLOW	0x00000040
#define NAK 17	0x00000080
#define NAK 8	0x00000100



Note • Any other Windows status may be used in the future, so mask away undefined bits in your application!

Windows APIs for Communication with the Printer

In order to make bi-directional communication easier and also compatible to more than one printer of the same kind on a specific PC, we implemented the Language Monitor function `GetPrinterData`. This is a Windows API described in the Windows documentation. To retrieve immediate printer status from the Spooler you can also use the function `GetPrinter`, however the `GetPrinterData` function is preferred over `GetPrinter` due to the fact that with `GetPrinterData`, all statuses and errors display, while with `GetPrinter`, only printer errors display.

GetPrinterData

The **GetPrinterData** function retrieves configuration data for the specified printer or print server.

Windows 2000/XP: Calling **GetPrinterData** is equivalent to calling the [GetPrinterDataEx](#) function with the *pKeyName* parameter set to "PrinterDriverData".

```
DWORD GetPrinterData(  
    HANDLE hPrinter, // handle to printer or print server  
    LPTSTR pValueName, // value name  
    LPDWORD pType, // data type  
    LPBYTE pData, // configuration data buffer  
    DWORD nSize, // size of configuration data buffer  
    LPDWORD pcbNeeded // bytes received or required  
);
```

Parameters

hPrinter

[in] Handle to the printer or print server for which the function retrieves configuration data. Use the `OpenPrinter` or `AddPrinter` function to retrieve a printer handle.

pValueName

[in] Pointer to a null-terminated string that identifies the data to retrieve.

For printers, this string is the name of a registry value under the printer's "PrinterDriverData" key in the registry.

For print servers, this string is one of the predefined strings listed in the following Remarks section.

pType

[out] Pointer to a variable that receives the type of data retrieved. The function returns the type specified in the `SetPrinterData` or `SetPrinterDataEx` call when the data was stored. This parameter can be `NULL` if you don't need the information.

pData

[out] Pointer to a buffer that receives the configuration data.

nSize

[in] Specifies the size, in bytes, of the buffer pointed to by *pData*.

pcbNeeded

[out] Pointer to a variable that receives the size, in bytes, of the configuration data. If the buffer size specified by *nSize* is too small, the function returns `ERROR_MORE_DATA`, and *pcbNeeded* indicates the required buffer size.

Return Values

If the function succeeds, the return value is `ERROR_SUCCESS`. If the function fails, the return value is an error value. Please see the standard Windows error codes in the Windows development documentation.

Remarks

`GetPrinterData` retrieves printer-configuration data set by the `SetPrinterDataEx` or `SetPrinterData` function.

If *hPrinter* is a handle to a print server, *pValueName* can specify one of the following predefined values.

`SPLREG_DEFAULT_SPOOL_DIRECTORY`
`SPLREG_PORT_THREAD_PRIORITY_DEFAULT`
`SPLREG_PORT_THREAD_PRIORITY`
`SPLREG_SCHEDULER_THREAD_PRIORITY_DEFAULT`
`SPLREG_SCHEDULER_THREAD_PRIORITY`
`SPLREG_BEEP_ENABLED`
`SPLREG_NET_POPUP`
`SPLREG_EVENT_LOG`
`SPLREG_MAJOR_VERSION`
`SPLREG_MINOR_VERSION`
`SPLREG_ARCHITECTURE`
`SPLREG_OS_VERSION`

Windows 2000/XP: `SPLREG_OS_VERSIONEX`

`SPLREG_DS_PRESENT` (On successful return, *pData* contains 0x0001 if the machine is on a DS domain, 0 otherwise.)

`SPLREG_DS_PRESENT_FOR_USER` (On successful return, *pData* contains 0x0001 if the user is logged onto a DS domain, 0 otherwise.)

`SPLREG_REMOTE_FAX` (On successful return, *pData* contains 0x0001 if the FAX service supports remote clients, 0 otherwise.)

SPLREG_NET_POPUP_TO_COMPUTER (On successful return, *pData* contains 1 if job notifications should be sent to the client computer, or 0 if job notifications are to be sent to the user.)

SPLREG_RETRY_POPUP (On successful return, *pData* contains 1 if server is set to retry PopUps for all jobs, or 0 if server does *not* retry PopUps for all jobs.)

In addition, the following values indicate pool printing behavior when an error occurs.

SPLREG_RESTART_JOB_ON_POOL_ERROR (Indicates the time, in seconds, when a job is restarted on another port after an error occurs. This is used with SPLREG_RESTART_JOB_ON_POOL_ENABLED.)

SPLREG_RESTART_JOB_ON_POOL_ENABLED (A nonzero value indicates that SPLREG_RESTART_JOB_ON_POOL_ERROR is enabled.)

The time specified in SPLREG_RESTART_JOB_ON_POOL_ERROR is a minimum time. The actual time can be longer, depending on the following port monitor settings, which are found under:

HKLM\SYSTEM\CurrentControlSet\Control\Print\Monitors\MonitorName\Ports.

Zebra has added status functionality in the Language Monitor function `GetPrinterDataFromPort` which is called by `GetPrinterData`. This way you can get printer-specific data through this Windows function.

The following table gives an overview of the Zebra specific keys (*pValueName*) used with `GetPrinterData`.

Table 8 • GetPrinterData Key Values

Printer DsMonitor Key	Explanation	Type
DeviceID	Printer's device ID string	REG_BINARY
ERROR	Printer Error or Status in Windows 16-bit format	REG_DWORD
ErrorEvent	Error event name for error event trigger	REG_SZ
EXTERNALERROR	Extended status according to Appendix B	REG_DWORD
Firmware	Firmware version	REG_BINARY
PAGECOUNT	Page counter for cut pages	REG_DWORD
PCB_REV	Printers PCB revision number	REG_BINARY
PCB_SN	Printers PCB serial number	REG_BINARY
StatusEvent	Status event name for status event trigger	REG_SZ
Monitor Key	Explanation	Type
ACK_SLEEP	ACK marker sleep time	REG_DWORD
READ_SLEEP	Sleep time before a read after write	REG_DWORD
READ_THREAD_SLEEP	Read thread sleep time	REG_DWORD
READ_REPEAT	Read repeat count.	REG_DWORD

For more information about the `ERROR` and `EXTERNALERROR` status response see the Remark section of the `GetPrinter` function.

GetPrinter

The **GetPrinter** function retrieves information about a specified printer.

```

BOOL GetPrinter(
    HANDLE hPrinter, // handle to printer
    DWORD Level, // information level
    LPBYTE *pPrinter, // printer information buffer
    DWORD cbBuf, // size of buffer
    LPDWORD pcbNeeded // bytes received or required
);

```

Parameters

hPrinter

[in] Handle to the printer for which the function retrieves information. Use the `OpenPrinter` or `AddPrinter` function to retrieve a printer handle.

Level

[in] Specifies the level or type of structure that the function stores into the buffer pointed to by *pPrinter*.

Windows NT/2000/XP: This value can be 1, 2, 3, 4, 5, 6, 7, 8 or 9.

pPrinter

[out] Pointer to a buffer that receives a structure containing information about the specified printer. The buffer must be large enough to receive the structure and any strings or other data to which the structure members point. If the buffer is too small, the *pcbNeeded* parameter returns the required buffer size.

The type of structure is determined by the value of *Level*.

Table 9 •

Level	Structure
1	A PRINTER_INFO_1 structure containing general printer information.
2	A PRINTER_INFO_2 structure containing detailed information about the printer.
3	Windows NT/2000/XP: A PRINTER_INFO_3 structure containing the printer's security information.
4	Windows NT/2000/XP: A PRINTER_INFO_4 structure containing minimal printer information, including the name of the printer, the name of the server, and whether the printer is remote or local.
5	A PRINTER_INFO_5 structure containing printer information such as printer attributes and time-out settings.
6	Windows 2000/XP: A PRINTER_INFO_6 structure specifying the status value of a printer.
7	Windows 2000/XP: A PRINTER_INFO_7 structure that indicates whether the printer is published in the directory service.
8	Windows 2000/XP: A PRINTER_INFO_8 structure specifying the global default printer settings.
9	Windows 2000/XP: A PRINTER_INFO_9 structure specifying the per-user default printer settings.

cbBuf

[in] Specifies the size, in bytes, of the buffer pointed to by *pPrinter*.

pcbNeeded

[out] Pointer to a variable that the function sets to the size, in bytes, of the printer information. If *cbBuf* is smaller than this value, **GetPrinter** fails, and the value represents the required buffer size. If *cbBuf* is equal to or greater than this value, **GetPrinter** succeeds, and the value represents the number of bytes stored in the buffer.

Return Values

If the function succeeds, the return value is a nonzero value.

If the function fails, the return value is zero. To get extended error information, call **GetLastError**.

Remarks

Zebra Printer status: It is recommended to use the [PRINTER_INFO_3](#) structure to inquire for the printer status presented by the LM.

Caution • The *pDevMode* member in the **PRINTER_INFO_2**, **PRINTER_INFO_8**, and **PRINTER_INFO_9** structures can be NULL. When this happens, the printer is unusable until the driver is reinstalled successfully.

Windows NT/2000/XP: For the [PRINTER_INFO_2](#) and [PRINTER_INFO_3](#) structures that contain a pointer to a security descriptor, the function retrieves only those components of the security descriptor that the caller has permission to read. To retrieve particular security descriptor components, you must specify the necessary access rights when you call the **OpenPrinter** function to retrieve a handle to the printer. The following table shows the access rights required to read the various security descriptor components.

Access Right	Security Descriptor Component
READ_CONTROL	Owner Primary group Discretionary access-control list (DACL)
ACCESS_SYSTEM_SECURITY	System access-control list (SACL)

Windows 2000/XP: If you specify level 7, the **dwAction** member of **PRINTER_INFO_7** returns one of the following values to indicate whether the printer is published in the directory service.

dwAction Value	Meaning
DSPRINT_PUBLISH	The printer is published. The pszObjectGUID member contains the GUID of the directory services print queue object associated with the printer.
DSPRINT_UNPUBLISH	The printer is not published.
DSPRINT_PENDING	Indicates that the system is attempting to complete a publish or unpublish operation. If a SetPrinter call fails to publish or unpublish a printer, the system makes further attempts to complete the operation in the background.

Zebra-added status functionality in the Language Monitor sets the spooler status with Windows defined status and a few additional status values.

The following tables give an overview of the Windows and Zebra status responses.

Windows Style Status Response

Table 10 • Windows Status Response

Printer Status		Windows Status	
Paper Jam	NAK 1	PRINTER_STATUS_PAPER_JAM	0x00000008
Cutter not home	NAK 2	PRINTER_STATUS_USER_INTERVENTION	0x00100000
Out of paper	NAK 3	PRINTER_STATUS_PAPER_OUT	0x00000010
Print head lifted	NAK 4	PRINTER_STATUS_DOOR_OPEN	0x00400000
Paper-feed error	NAK 5	PRINTER_STATUS_PAPER_PROBLEM	0x00000040
Temperature error	NAK 6	PRINTER_STATUS_ERROR	0x00000002
Presenter jam	NAK 7	PRINTER_STATUS_ERROR	0x00000002
Paper jam during retract	NAK 8	PRINTER_STATUS_ERROR	0x00000002
Index Error	NAK 12	PRINTER_STATUS_ERROR	0x00000002
Checksum error	NAK 13	PRINTER_STATUS_ERROR	0x00000002
Wrong FW	NAK 14	PRINTER_STATUS_ERROR	0x00000002
Retract occurred	NAK 16	PRINTER_STATUS_ERROR	0x00000002
Paused	NAK 17	PRINTER_STATUS_ERROR	0x00000002
Paper near end	0x00 0x02	PRINTER_STATUS_PAPER_NEAR_END	0x02000000
Weekend sensor	0x00 0x40	PRINTER_STATUS_PAPER_WEEKEND	0x04000000
Paper at presenter	0x00 0x08	PRINTER_STATUS_PAPER_PRESENTER	0x08000000
Buffer overflow	0x10 0x00	PRINTER_STATUS_ERROR	0x00000002
Print data exist	0x40 0x00	PRINTER_STATUS_PROCESSING	0x00004000
External error		PRINTER_STATUS_ERROR	0x00000002



Note • The spooler status is changed by SetPort. When using SetPort with custom messages, you can't set these to be displayed or used by the spooler. This is a known bug; SetPort doesn't work with custom status messages. (Microsoft) Therefore, all custom messages have been exchanged to PRINTER_STATUS_ERROR. The custom messages are only accessible through the GetPrinterData function.

Extended Error Status Response

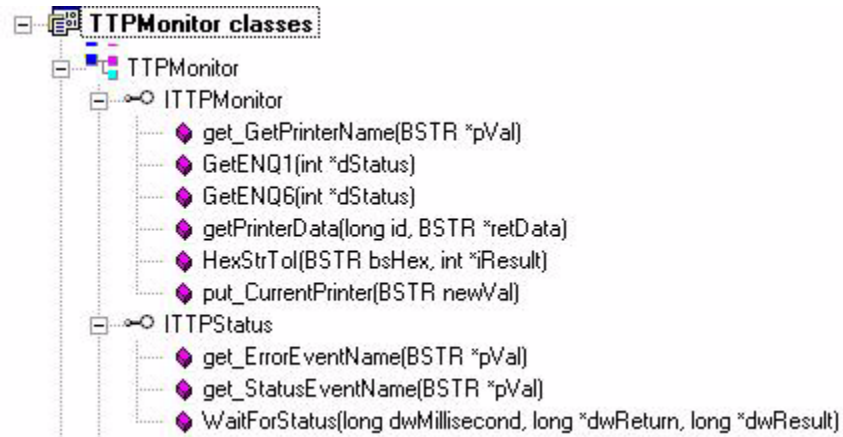
Table 11 • Extended Status Response

Printer Status		Decode With
Temperature error	NAK 6	0x00000001
Presenter jam	NAK 7	0x00000002
Index Error	NAK 12	0x00000004
Checksum error	NAK 13	0x00000008
Wrong FW	NAK 14	0x00000010
Retract occurred	NAK 16	0x00000020
Buffer overflow		0x00000040
Paused	NAK 17	0x00000080
Paper jam during retract	NAK 8	0x00000100

ATL Object for Communication with the Printer

In order to make implementation of the original LmPrinterIoControl function easier, Zebra implemented an ATL object TTPMONITOR. This DLL provides a function interface which is easily accessible from many programming languages. It is not recommended for use in new applications. For new applications, use the Windows API functionality as described above.

ATL Object Function and Property Definition



The SetCurrentPrinter property sets the current printer and loads the LM DLL and initializes the Error and Status events.

- get_GetPrinterName will return the set printer name.
- getPrinterData returns the status information from the LmPrinterIoControl function.

The parameter id can have one of the following values:

Table 12 • getPrinterData ID Values

#define	TTPKEY_LANGUAGE_MONITOR	0
#define	TTPKEY_STATUS_GENERAL	1
#define	TTPKEY_PAPER_NEAR_END	2
#define	TTPKEY_STATUS_REPORT	3
#define	TTPKEY_PROGRAM_VERSION	4
#define	TTPKEY_SERIAL_NUMBER	5
#define	TTPKEY_HARDWARE_REVISION	6
#define	TTPKEY_DEVICE_ID	7
#define	TTPKEY_GETPARAMETER	8
#define	TTPKEY_AUTO_STATUS	9
#define	TTPKEY_EXT_AUTO_STATUS	10
#define	TTPKEY_DEVICE_ID_FC	11

Table 12 • getPrinterData ID Values (Continued)

#define	TTPKEY_RESET_FULL	12
#define	TTPKEY_RESET_INITIALIZE	13

- GetENQ1 returns only the printer status ESC ENQ 1 in a WORD, in the form second byte and then first byte.
- GetENQ6 returns only the printer status ESC ENQ 6 in a WORD, in the form second byte and then first byte.
- GetErrorEventName returns the “Error event name” after the call of SetCurrentPrinter.
- GetStatusEventName returns the “Status event name” after the call of SetCurrentPrinter.
- HexStrToI implements a conversion function for the getPrinterData function results which are Hex string results. The result of this function is an Integer representation of the Hex string value.
- WaitForStatus implements a WaitForMultipleObjects and returns with the result of the Wait function and the status result from the LmPrinterIoControl function call.

Event Notification

For a program to not have to continuously poll the printer for status, Zebra implemented an Event notification in the Language Monitor.

This notification, used together with the WaitForStatus function in the ATL object or the WaitForMultipleObjects Windows function, enables applications to react on status changes rather than looking for status periodically.

When the internal polling thread recognizes a status change or error then it will fire an event, either an error or a status event.

The Application or the ATL object (TTPMONITOR) can open an event object to the LM events and initialize a “Wait for event” function. The necessary event names can be extracted from the registry.

When an event occurs, call the GetPrinterData function to get the error or status condition in the DWORD returned.



Note • To extract registry information you need to:

1. Starting with the printer name, open the HKEY_LOCAL_MACHINE with the following path: "SYSTEM\\CurrentControlSet\\Control\\Print\\Printers\\%s" where %s stands for the printer name.
2. Extract the string “Printer Driver” from this key.
3. Open the Printer driver with the following path:
"HKLM\\SYSTEM\\CurrentControlSet\\Control\\Print\\Environments\\Windows NT x86\\Drivers\\Version-3\\%s" where %s is the extracted printer driver from step 2.
4. Extract the string “Monitor” from this key
5. Open the Monitor with the following path:
"HKLM\\SYSTEM\\CurrentControlSet\\Control\\Print\\Monitors\\%s" where %s is the extracted monitor from step 4.
6. Extract the string “Driver” (the LM DLL) to be opened for function calls
7. Extract the Event names with the strings “ErrorEvent” and “StatusEvent” in older drivers or use step 8 to open a new Key in newer drivers (after September 2004).
8. Open the Printer with the following path:
"HKLM\\SYSTEM\\CurrentControlSet\\Control\\Print\\Printers\\%s\\PrinterDriverData"
and extract the event names with the strings “ErrorEvent” and “StatusEvent”.

The recommended way to extract the Event names is to call the GetPrinterData function with ErrorEvent or StatusEvent as the key value and extract the event name to open the event handle.

Registry Entries

In the Language Monitor Key

ACK_SLEEP = REG_DWORD	00000064	(100 decimal)
DeleteJob = REG_DWORD	00000001	(1 decimal)
Driver = REG_SZ	"10x0MON.DLL" or the equivalent LM for the specific printer	
READ_REPEAT = REG_DWORD	00000001	(1 decimal)
READ_SLEEP = REG_DWORD	00000064	(100 decimal)
READ_THREAT_SLEEP = REG_DWORD	000005dc	(1500 decimal)
TransmissionRetryTimeout = REG_SZ	5	

For older drivers (before September 2004) the following two entries are still available.

ErrorEvent = REG_SZ	"ErrorEvent1" or the equivalent name for the specific printer
StatusEvent = REG_SZ	"StatusEvent1" or the equivalent name for the specific printer

- The READ_THREAD_SLEEP controls the auto status inquiry time.

When the LM initializes it starts a read thread which in turn runs until the LM closes down.

During print idle time, the read thread is inquiring the status from the printer and signals a status change to the listening application.

- ACK_SLEEP controls the sleep time in case of a page hold inquiry.

The page hold function is not available for all printers and is used together with a driver setting in the 8x00 series driver.

- READ_SLEEP controls the sleep time between a status inquiry and the successive read call.

When the LM is inquiring for status the printer needs time to gather the needed information and therefore can't react immediately. Since some printers react faster than others and the speed of the PC system and the used OS are also variables which differ, this key was implemented to fine-tune the application.

- READ_REPEAT controls the amount of re-tries on a failed read inside the LM read function.
- DeleteJob controls the LM behavior when an error appears. In the case this value is 1 the LM will delete all print jobs and if it is 0 it will hold the job and stop any spooler activities until the error is resolved.
- TransmissionRetryTimeout is a parallel timeout setting in the Printer Properties which can cause long delays while waiting for status. The Language Monitor is setting this to the value in the registry key.

In the Printer Key

In recent drivers (after September 2004) the Error and Status event names have been moved into a separate key (PrinterDriverData) in the Printers section. This is to allow multiple printers of the same kind to reflect status and error conditions independently, on the same PC.

DeviceID = REG_SZ	the printer's DeviceID in hex value
ERROR = REG_BINARY	the printer's error status according to Windows status values
EXTERNALERROR = REG_BINARY	the printer's error status according to ESC ENQ 1 (Zebra-specific)
ErrorEvent = REG_SZ	"ErrorEvent1" or the equivalent name for the specific printer
PAGECOUNT = REG_DWORD	current page count
PCB_REV = REG_BINARY	printer's PCB revision
PCB_SN = REG_BINARY	printer's serial number
StatusEvent = REG_SZ	"StatusEvent1" or the equivalent name for the specific printer

ErrorEvent and StatusEvent are the event names used in the LM to signal the current status changes. When you open an event with the listed name you can wait for this event to happen and inquire with the "Auto Status" or "Ext Auto Status" for the status value.



Note • The difference between "Auto Status" and "Ext Auto Status" is the return value. In case of "Auto Status" you will get a DWORD back that is also stored in the ERROR value in the registry and in case "Ext Auto Status" you will get two DWORD's back where the first is the ERROR and the second the EXTERNALERROR value in the registry which reflects the printers ESC ENQ 1 value.



Note • When changes have been made in the registry it is necessary to reboot the PC (or restart the spooler service) in order for the changed parameters to take effect.



Status Messages

Below statuses are defined in winspool.h.

Table 13 • Winspool Statuses

#define PRINTER_STATUS_ERROR	0x00000002
#define PRINTER_STATUS_PAPER_JAM	0x00000008
#define PRINTER_STATUS_PAPER_OUT	0x00000010
#define PRINTER_STATUS_PAPER_PROBLEM	0x00000040
#define PRINTER_STATUS_OFFLINE	0x00000080
#define PRINTER_STATUS_USER_INTERVENTION	0x00100000
#define PRINTER_STATUS_DOOR_OPEN	0x00400000

Below statuses are defined by Zebra.

Table 14 • Zebra Statuses

#define PRINTER_STATUS_PAPER_NEAR_END	0x02000000
#define PRINTER_STATUS_PAPER_WEEKEND	0x04000000
#define PRINTER_STATUS_PAPER_AT_PRESENTER	0x08000000
#define PRINTER_STATUS_EXTERNAL_ERROR	0x10000000



Notes • _____



External Error Status Messages

Below statuses are External Error statuses defined by Zebra.

Table 15 • External Error Statuses

#define NAK6	0x00000001
#define NAK7	0x00000002
#define NAK12	0x00000004
#define NAK13	0x00000008
#define NAK14	0x00000010
#define NAK16	0x00000020
#define BUFFEROVERFLOW	0x00000040
#define NAK 17	0x00000080
#define NAK 8	0x00000100



Notes • _____



Programming Examples

Background

In order to incorporate the new way of status monitoring you need to get a little bit of background on what happens in a Kiosk when you print and when you should monitor your status.

Status monitoring can be handled in two different ways.

- Monitor in the printing application
- Monitor in a separate application

When you monitor in your printing application you would commonly look at the printer before sending a print job to see if the printer is OK and then send your print job. After the print job is signaled as being printed you would check status again to see if the printer has any errors or if the paper has been taken, etc.

Monitoring in a separate application usually doesn't allow direct interaction with the printed job so you are trying to poll the printer as often as you can to get most accurate information on what the printer is doing. This is usually a very time consuming task and you have to care for such things as not to interfere with a current print job.

Since the latter example is most commonly used for status monitoring, we have incorporated the even notification into the Language Monitor to allow a monitoring application to do other tasks and have a separate thread listening for the printer status or error event change. When this occurs the thread is simply getting the status and reporting this back to the main program or doing any other kind of reporting.

To accommodate this notification for all error and status changes we incorporated two mechanisms in the LM.

Monitoring while printing

We implemented status monitoring in the internal printing structure of the Language Monitor. When you Open a Document, print it and close the Document again the LM will check the printer status before and after printing and will also react to write errors if such occur. Then it will set the printer status and raise the error event.

Monitoring while idle

We implemented an internal status thread which polls the printer when it is idle in a predefined cycle and provides changed status information in the same manner. It will set the status and raise an error or status event. Therefore, it is not necessary to implement your own monitoring loop. You can simply wait for an event in your application's idle loop.

Implementation

The sample program used to demonstrate the status monitoring with GetPrinterData is Monitor2 and the main dialog "Monitor2Dlg.cpp" implements the necessary steps.

Open the Printer

The first step of your implementation is to open the printer you want to monitor and get the Error event and Status even name. You can find this in the BOOL CMonitor2Dlg::OnInitDialog() function.

```
bRet = OpenPrinter(m_csPrinter.GetBuffer(1), &hPrinter,
    &pd);
...
if ((dRet = GetPrinterData(hPrinter, "ErrorEventName",
    &dType, (LPBYTE)cTmp, 100, &dNeeded))!=ERROR_SUCCESS)
...
if ((dRet = GetPrinterData(hPrinter, "StatusEventName",
    &dType, (LPBYTE)cTmp, 100, &dNeeded))!=ERROR_SUCCESS)
...

```

Open the Event Handles

Then you open the two event handles and fill these handles into a structure you will pass on to the new thread. You can find this in the BOOL CMonitor2Dlg::OnInitDialog() function.

```
typedef struct _CStatusThreadInfo
{
    HWNDmyHwnd;
    DWORDdSleepTime;
    HANDLEhPrinter;
    HANDLEhError;
    HANDLEhStatus;
    BOOLm_hStatusEventKillThread;
} CStatusThreadInfo;
...
if ((cTi.hError = OpenEvent(SYNCHRONIZE, TRUE,
    m_csErrorEvent))==NULL)
...
if ((cTi.hStatus = OpenEvent(SYNCHRONIZE, TRUE,
    m_csStatusEvent))==NULL)

```

Start Monitoring

When all this is done you can start your monitoring thread. You can find this in the void `CMonitor2Dlg::OnAutoStatus()` function.

```
m_StatusThread = AfxBeginThread( StatusThreadProc, &cTi,  
    THREAD_PRIORITY_NORMAL, 0, 0, NULL );
```

Fill Event Arrays

In the monitoring thread you create and fill an array of handles with the error and status event handle. You can find this in the `UINT StatusThreadProc(LPVOID pParam)` function.

```
myHandle[0] = pInfo->hError;  
myHandle[1] = pInfo->hStatus;
```


Start the Waiting Loop

Then you are ready to start the waiting loop which you find in the `UINT StatusThreadProc(LPVOID pParam)` function.

```
for ( ; ; )
{
    if (pInfo->m_hStatusEventKillThread)
    {
        OutputDebugStringA("### [Thread msg.] Kill
thread...\n");
        pInfo->m_hStatusEventKillThread = FALSE;
        AfxEndThread( 1 );
        return 1;
    }
    if ((dwRet = WaitForMultipleObjects(2, myHandle, FALSE,
pInfo->dSleepTime))!=WAIT_FAILED)
    {
        if (dwRet==WAIT_OBJECT_0 || dwRet==WAIT_OBJECT_0+1)
        {
            if ((dwRet = GetPrinterData(hPrinter, "ERROR",
&dType, (LPBYTE)&dwResult, sizeof(dwResult),
&dNeeded))!=ERROR_SUCCESS)
            {
                sprintf( str, "### [Status Thread error %d]
read [%08X]\n", dwRet, dwResult);
                OutputDebugStringA(str);
            }
            sprintf( str, "### [Status Thread] read
[%08X]\n", dwResult);
            OutputDebugStringA(str);
            SendMessage(GetDlgItem((HWND)pInfo->myHwnd,
IDC_Status), WM_SETTEXT, 0, (LPARAM)(str));
            if (dwResult & 0x00000000)
                SendMessage(GetDlgItem((HWND)pInfo->myHwnd,
IDC_Status), WM_SETTEXT, 0,
(LPARAM)("PRINTER_STATUS_OK"));
            if (dwResult & PRINTER_STATUS_ERROR)
                SendMessage(GetDlgItem((HWND)pInfo->myHwnd,
IDC_Status), WM_SETTEXT, 0,
(LPARAM)("PRINTER_STATUS_ERROR"));
            if (dwResult & PRINTER_STATUS_PENDING_DELETION)
                SendMessage(GetDlgItem((HWND)pInfo->myHwnd,
IDC_Status), WM_SETTEXT, 0,
```

```

        (LPARAM) ("PRINTER_STATUS_PENDING_DELETION" ));
if (dwResult & PRINTER_STATUS_PAPER_JAM)
    SendMessage(GetDlgItem((HWND)pInfo->myHwnd,
        IDC_Status), WM_SETTEXT, 0,
        (LPARAM) ("PRINTER_STATUS_PAPER_JAM" ));
if (dwResult & PRINTER_STATUS_PAPER_OUT)
    SendMessage(GetDlgItem((HWND)pInfo->myHwnd,
        IDC_Status), WM_SETTEXT, 0,
        (LPARAM) ("PRINTER_STATUS_PAPER_OUT" ));
if (dwResult & PRINTER_STATUS_PAPER_PROBLEM)
    SendMessage(GetDlgItem((HWND)pInfo->myHwnd,
        IDC_Status), WM_SETTEXT, 0,
        (LPARAM) ("PRINTER_STATUS_PAPER_PROBLEM" ));
if (dwResult & PRINTER_STATUS_OFFLINE)
    SendMessage(GetDlgItem((HWND)pInfo->myHwnd,
        IDC_Status), WM_SETTEXT, 0,
        (LPARAM) ("PRINTER_STATUS_OFFLINE" ));
if (dwResult & PRINTER_STATUS_IO_ACTIVE)
    SendMessage(GetDlgItem((HWND)pInfo->myHwnd,
        IDC_Status), WM_SETTEXT, 0,
        (LPARAM) ("PRINTER_STATUS_IO_ACTIVE" ));
if (dwResult & PRINTER_STATUS_BUSY)
    SendMessage(GetDlgItem((HWND)pInfo->myHwnd,
        IDC_Status), WM_SETTEXT, 0,
        (LPARAM) ("PRINTER_STATUS_BUSY" ));
if (dwResult & PRINTER_STATUS_PRINTING)
    SendMessage(GetDlgItem((HWND)pInfo->myHwnd,
        IDC_Status), WM_SETTEXT, 0,
        (LPARAM) ("PRINTER_STATUS_PRINTING" ));
if (dwResult & PRINTER_STATUS_OUTPUT_BIN_FULL)
    SendMessage(GetDlgItem((HWND)pInfo->myHwnd,
        IDC_Status), WM_SETTEXT, 0,
        (LPARAM) ("PRINTER_STATUS_OUTPUT_BIN_FULL" ));
if (dwResult & PRINTER_STATUS_PROCESSING)
    SendMessage(GetDlgItem((HWND)pInfo->myHwnd,
        IDC_Status), WM_SETTEXT, 0,
        (LPARAM) ("PRINTER_STATUS_PROCESSING" ));
if (dwResult & PRINTER_STATUS_USER_INTERVENTION)
    SendMessage(GetDlgItem((HWND)pInfo->myHwnd,
        IDC_Status), WM_SETTEXT, 0,
        (LPARAM) ("PRINTER_STATUS_USER_INTERVENTION" ));
if (dwResult & PRINTER_STATUS_DOOR_OPEN)
    SendMessage(GetDlgItem((HWND)pInfo->myHwnd,
        IDC_Status), WM_SETTEXT, 0,
        (LPARAM) ("PRINTER_STATUS_DOOR_OPEN" ));
if (dwResult & PRINTER_STATUS_PAPER_NEAR_END)

```

```

        SendMessage(GetDlgItem((HWND)pInfo->myHwnd,
        IDC_Status), WM_SETTEXT, 0,
        (LPARAM)("PRINTER_STATUS_PAPER_NEAR_END"));
    if (dwResult & PRINTER_STATUS_PAPER_WEEKEND)
        SendMessage(GetDlgItem((HWND)pInfo->myHwnd,
        IDC_Status), WM_SETTEXT, 0,
        (LPARAM)("PRINTER_STATUS_PAPER_WEEKEND"));
    if (dwResult & PRINTER_STATUS_PAPER_PRESENTER)
        SendMessage(GetDlgItem((HWND)pInfo->myHwnd,
        IDC_Status), WM_SETTEXT, 0,
        (LPARAM)("PRINTER_STATUS_PAPER_PRESENTER"));
    if (dwResult & PRINTER_STATUS_EXTERNAL_ERROR)
    {
        SendMessage(GetDlgItem((HWND)pInfo->myHwnd,
        IDC_Status), WM_SETTEXT, 0,
        (LPARAM)("PRINTER_STATUS_EXTERNAL_ERROR"));
        if ((dwRet = GetPrinterData(hPrinter,
        "EXTERNALERROR", &dType, (LPBYTE)dwResult,
        sizeof(dwResult), &dNeeded))!=ERROR_SUCCESS)
        {
            sprintf( str, "### [Status Thread error %d]
            read [%08X]\n", dwRet, dwResult);
            OutputDebugStringA(str);
        }
        sprintf( str, "### [Status Thread External
        Error] read [%08X]\n", dwResult);
        OutputDebugStringA(str);
        SendMessage(GetDlgItem((HWND)pInfo->myHwnd,
        IDC_Status), WM_SETTEXT, 0, (LPARAM)(str));
    }
}
else
    SendMessage(GetDlgItem((HWND)pInfo->myHwnd,
    IDC_Status), WM_SETTEXT, 0, (LPARAM)("Timeout"));
}
else
{
    dwRet = GetLastError();
    sprintf( str, "### Wait function failed! [%d]\n",
    dwRet);
    OutputDebugStringA(str);
}
}
}

```

When an event occurs you need to get the status with `GetPrinterData` and decode the result according to the sample or any way you feel necessary. In any case you can send a message or do any form of status reporting you want to do.

Index



A

- advanced, driver properties 31
- after print, clear presenter 16
- after print, hold page 17
- appendix
 - A 49
 - B 51, 53
- atl object
 - communication with printer 44
 - function 44
 - property definition 44
- auto status
 - windows compatible status 34

B

- before print 19
- bidirectional support
 - enable 24
- bits
 - data 25
 - per second 25
 - stop 25

C

- cartridges. See <Emphasis>media cartridges
- cbbuf 41
- clear, all print jobs
 - presenter jam 31
- commands
 - printer 18
- communication
 - with printer, windows API's 36

- communication with printer
 - atl object 44
 - zebra functions 34
- configure, port 25
- contacts 6
- control
 - flow 25
- copy count 10
- count, copy 10
- customer service 6
- cut mode, paper 16
- cut per document
 - page hold enabled 31
- cut/eject
 - restarting print job 31

D

- data bits 25
- device settings 21
- document options 11
- driver properties, advanced 31

E

- eject, paper 18
- enable, bidirectional support 24
- entries, registry 47
- event notification 46
- ext auto status
 - windows compatible status 35
- extended error
 - status response 43

F

features, printer 12
flow control 25

G

getprinter 39
 parameters 40
 cbbuf 41
 hprinter 40
 level 40
 pcbneeded 41
 pprinter 40
getprinterdata 36

H

hold page, after print 17
hprinter 36, 40

K

keys
 printer 28

L

language monitor 31, 33
level 40
liability 2

M

media
 ordering 6
media cartridges
 ordering 6
media ordering 6
mode, paper cut 16
mount, vertical 17

N

notification, event 46
nsize 37

O

offline/online status
 not automatically updated 32
online/offline status 32
options
 document 11
ordering media 6
ordering media cartridges 6

ordering ribbon and media 6
output, paper 9

P

page hold enabled
 retract behavior
 set to eject 31
 using cut per document 31
page type 13
paper
 cut mode 16
 eject 18
 output 9
 sensitivity 14
 size 9
paper output 9
paper size 9
parameters 36
 hprinter 36
 nsize 37
 pcbneeded 37
 pdata 37
 ptype 36
 pvaluenam 36
parameters, printer 12
parity 25
pcbneeded 37, 41
pdata 37
port
 configure 25
ports 23
pprinter 40
preferences, printing 7
presenter
 clear after print 16
presenter jam
 clear all print jobs 31
printer
 commands 18
 features 12
 keys 28
 parameters 12, 18
printing preferences 7
printing speed 14
ptype 36
pvaluenam 36

R

registry entries 47
 language monitor key 47
 printer key 48

- remarks 37, 41
- restarting
 - print job
 - cut/eject 31
- restrictions
 - advanced driver properties 31
 - using language monitor 31
- retract behavior
 - set to eject
 - page hold enabled 31
- retract setting 15
- return values 37, 41
- reverse 19
- reverse, before print 19
- ribbon ordering 6

S

- sales 6
- sensitivity, paper 14
- setting
 - retract 15
- settings
 - device 21
- size, paper 9
- speed, printing 14
- status
 - defined by Zebra 35
 - defined in winspool.h 34

- status response
 - extended error 43
 - windows style 42
- stop bits 25

T

- technical support 6
- type, page 13

V

- values
 - return 37, 41
- vertical mount 17

W

- windows API's
 - communication with printer 36
- windows compatible status
 - auto status 34
 - ext auto status 35
- windows style
 - status response 42
- winspool.h
 - statuses defined in 34

Z

- zebra functions
 - communication with printer 34



Zebra Technologies International, LLC

333 Corporate Woods Parkway
Vernon Hills, Illinois 60061.3109 U.S.A
T: +1 847 793 2600
Toll-free +1 800 423 0422
F: +1 847 913 8766

Zebra Technologies Europe Limited

Dukes Meadow
Millboard Road
Bourne End
Buckinghamshire, SL8 5XF, UK
T: +44 (0)1628 556000
F: +44 (0)1628 556001

Zebra Technologies Asia Pacific, LLC

120 Robinson Road
#06-01 Parakou Building
Singapore 068913
T: +65 6858 0722
F: +65 6885 0838

<http://www.zebra.com>