

# Zebra® Kiosk OPOS Driver

---

## User Guide



**© 2016 ZIH Corp.** The copyrights in this manual, software and/or firmware in the printer described therein are owned by ZIH Corp. and Zebra's licensor's. 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<sup>®</sup>, ZPL II<sup>®</sup>, and ZebraLink<sup>™</sup> programs; Element Energy Equalizer<sup>™</sup> Circuit; E<sup>3</sup><sup>™</sup>; and Monotype Imaging fonts. Software © ZIH Corp. All rights reserved worldwide.

ZebraLink, Element Energy Equalizer, E<sup>3</sup> and all product names and numbers are trademarks, and Zebra, the Zebra head graphic, ZPL and ZPL II 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

Who Should Use This Document .....	3
Core Companies .....	4
Document Conventions .....	5
Kiosk OPOS Driver Installation .....	7
Hardware Requirements .....	7
Compatible Operating Systems .....	7
Before Installing the OPOS Driver .....	8
Installing the OPOS Driver .....	8
Testing the Installation .....	13
OLE for Retail POS Controls .....	15
How an Application Uses an OPOS Control .....	16
Device Power Reporting Model .....	16
Summary of Common Properties, Methods, and Events .....	17
DirectIO Method .....	19
Events .....	21
POS Printer Summary .....	21
General Information .....	28
Data Characters and Escape Sequences .....	29

Properties .....	33
CapCharacterSet Property .....	33
CapRecMarkFeed Property .....	33
CapRecNearEndSensor Property .....	33
CapRecPapercut Property .....	33
CharacterSet Property .....	33
CharacterSetList Property .....	34
FontTypefaceList Property .....	34
MapMode Property .....	34
RecBarcodeRotationList Property .....	34
RecLineChars Property .....	34
RecLineCharsList Property .....	34
RecLineHeight Property .....	34
RecLineSpacing Property .....	35
RecLinesToPaperCut Property .....	35
RecLineWidth Property .....	35
RecSidewaysMaxChars Property .....	35
RecSidewaysMaxLines Property .....	35
RotateSpecial Property .....	35
Methods .....	36
CutPaper Method .....	36
PrintBarcode Method .....	36
PrintBitmap Method .....	38
RotatePrint Method .....	38
SetBitmap Method .....	39

# About This Document

This section provides you with contact information, additional references, and document conventions.

## Who Should Use This Document

The Kiosk OPOS Driver User Guide is targeted to an application developer who requires access to POS-specific peripheral devices.

## OPOS Information

Refer to the following websites for OPOS information:

- Reference implementation – Common Control Objects:  
<http://monroecs.com/opos.htm>
- NRF-ARTS Standards Body:  
<http://www.nrf-arts.org>

Information in this document is subject to change without notice.

- © 1995-2000 Fujitsu-ICL. All rights reserved.
- © 1995-2000 Microsoft Corporation. All rights reserved.
- © 1995-2000 NCR Corporation. All rights reserved.
- © 1995-2000 Seiko Epson Corporation. All rights reserved.

## OLE for Retail POS Committee

### Core Companies

- Epson
- Fujitsu-ICL
- Microsoft
- NCR
- plus—
- OPOS-Japan
- OPOS-Europe

## Document Conventions

The following conventions are used throughout this document to convey certain information.

**Alternate Color** (online only) Cross-references contain hot links to other sections in this guide. If you are viewing this guide online in .pdf format, you can click the cross-reference ([blue text](#)) to jump directly to its location.

### Icons Used



**Important** • Advises you of information that is essential to complete a task.



**Note** • Indicates neutral or positive information that emphasizes or supplements important points of the main text.





# Kiosk OPOS Driver Installation

## Kiosk OPOS Driver Installation

The following sections describe:

- *Hardware Requirements* [on page 7](#)
- *Compatible Operating Systems* [on page 7](#)
- *Before Installing the OPOS Driver* [on page 8](#)
- *Installing the OPOS Driver* [on page 8](#)
- *Testing the Installation* [on page 13](#)

### Hardware Requirements

- Intel® Pentium® P2 processor or equivalent
- 128 Mb RAM
- 150 Mb Hard Disk space
- Compliant USB port

### Compatible Operating Systems

- Windows 8 (32 bit only)
- Windows 7 (32 bit only)
- Windows XP
- Windows 2003 Server
- Windows 2000

## Before Installing the OPOS Driver

Before you start installing the OPOS Driver, install the appropriate Kiosk Printer Driver and power on your printer.

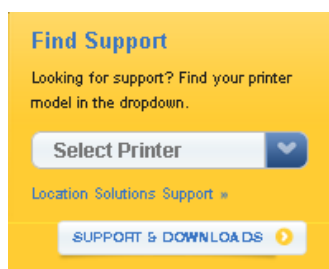
Drivers	Firmware & Service Packs	Software Utilities	Manuals
Recommended	Version	Operating System	Download
Kiosk OPOS Driver ( <a href="#">View Release Notes</a> )	v1.11.0.23	Windows 7, Windows Vista, Windows XP	<a href="#">Download (2 MB)</a> »
Kiosk Printer Driver ( <a href="#">View Release Notes</a> )	v1.3.510.83	Windows 7, Windows Vista, Windows 2003, Windows XP	<a href="#">Download (3 MB)</a> »
TTP 2030 EC7 Driver ARMv5 ( <a href="#">View Release Notes</a> )	0.3.512.3	Windows EC 7	<a href="#">Download (283 KB)</a> »
TTP 2030 EC7 Driver x86 ( <a href="#">View Release Notes</a> )	0.3.512.3	Windows EC 7	<a href="#">Download (263 KB)</a> »
TTP 2030 CE6 Driver ARMv4 ( <a href="#">View Release Notes</a> )	0.3.512.3	Windows CE 6.0	<a href="#">Download (223 KB)</a> »
TTP 2030 CE6 Driver x86 ( <a href="#">View Release Notes</a> )	0.3.512.3	Windows CE 6.0	<a href="#">Download (212 KB)</a> »

## Installing the OPOS Driver

After you install the appropriate Kiosk Printer Driver, you can install the Kiosk OPOS Driver.

### To install the Kiosk OPOS Driver

1. Go to [www.zebra.com](http://www.zebra.com).



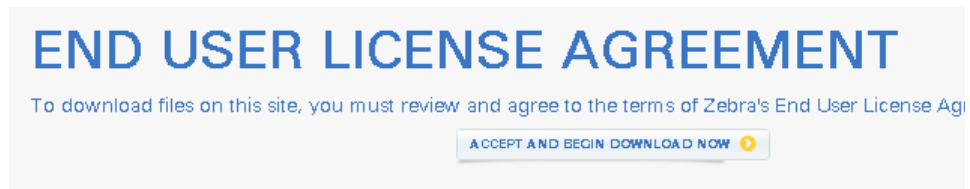
2. In the **Find Support** list, select your printer.
3. Click the **Drivers** tab.

The list of available drivers appear.

Drivers	Firmware & Service Packs	Software Utilities	Manuals
Recommended	Version	Operating System	Download
Kiosk OPOS Driver ( <a href="#">View Release Notes</a> )	v1.11.0.23	Windows 7, Windows Vista, Windows XP	<a href="#">Download (2 MB)</a> »
Kiosk Printer Driver ( <a href="#">View Release Notes</a> )	v1.3.510.83	Windows 7, Windows Vista, Windows 2003, Windows XP	<a href="#">Download (3 MB)</a> »
TTP 2030 EC7 Driver ARMv5 ( <a href="#">View Release Notes</a> )	0.3.512.3	Windows EC 7	<a href="#">Download (283 KB)</a> »
TTP 2030 EC7 Driver x86 ( <a href="#">View Release Notes</a> )	0.3.512.3	Windows EC 7	<a href="#">Download (263 KB)</a> »
TTP 2030 CE6 Driver ARMv4 ( <a href="#">View Release Notes</a> )	0.3.512.3	Windows CE 6.0	<a href="#">Download (223 KB)</a> »
TTP 2030 CE6 Driver x86 ( <a href="#">View Release Notes</a> )	0.3.512.3	Windows CE 6.0	<a href="#">Download (212 KB)</a> »

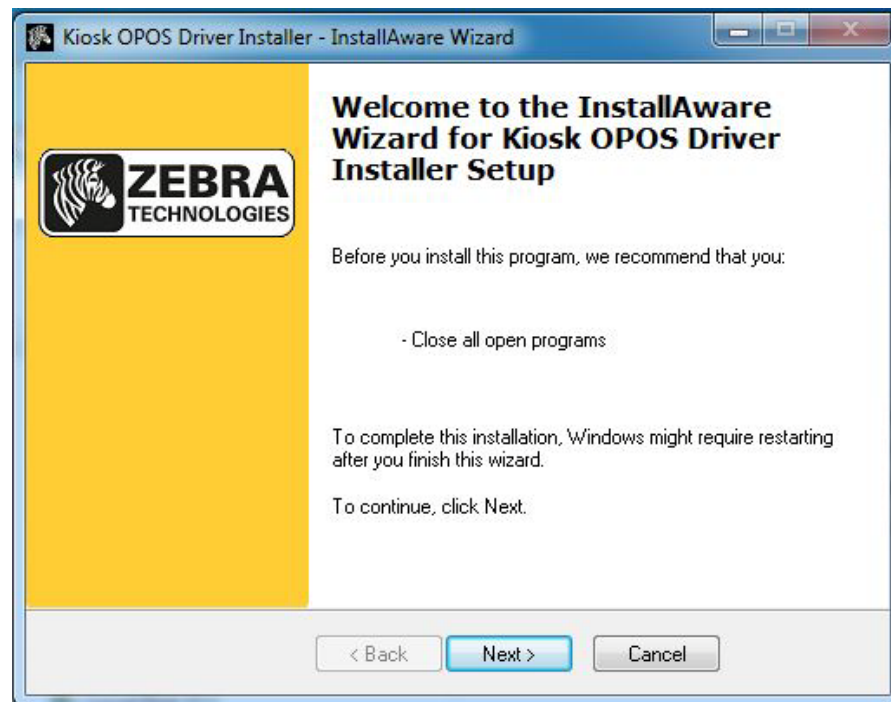
4. In the Kiosk OPOS Driver row, click **Download**.

The End User License Agreement appears.



5. Click **Accept and Begin Download Now**, and save the **zebra-kiosk-opos-driver-installer 1-11-1-2.exe** file.
6. Double-click **zebra-kiosk-opos-driver-installer 1-11-1-2.exe** to start the installer.  
A message appears asking if you want to allow the program to make changes to the computer.
7. Click **Yes**.

The Welcome screen appears.



**8. Click Next.**

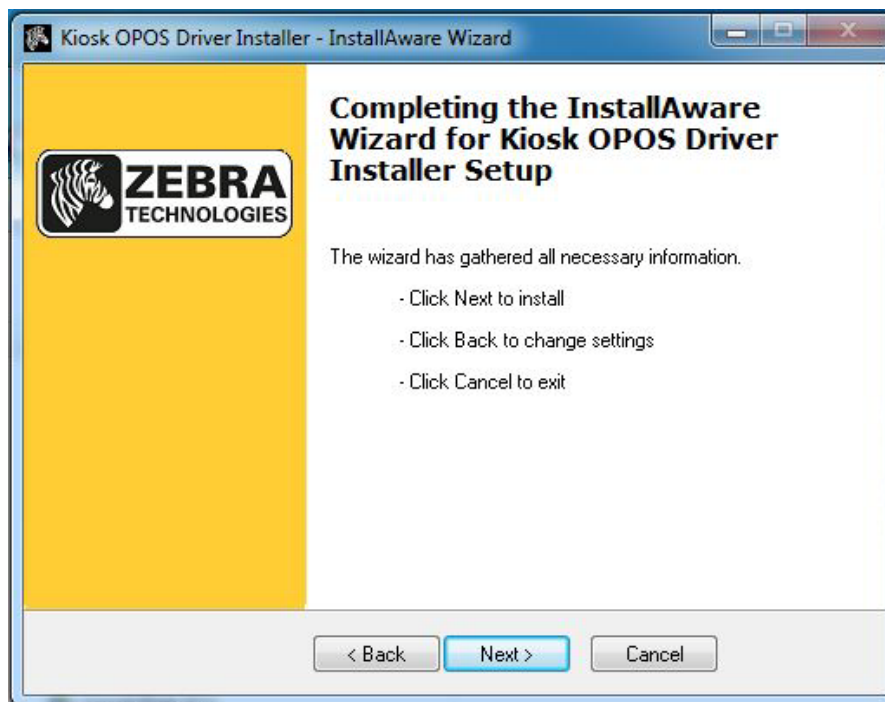
The License Agreement appears.



**9. Click the I Agree check box, and then click Next.**

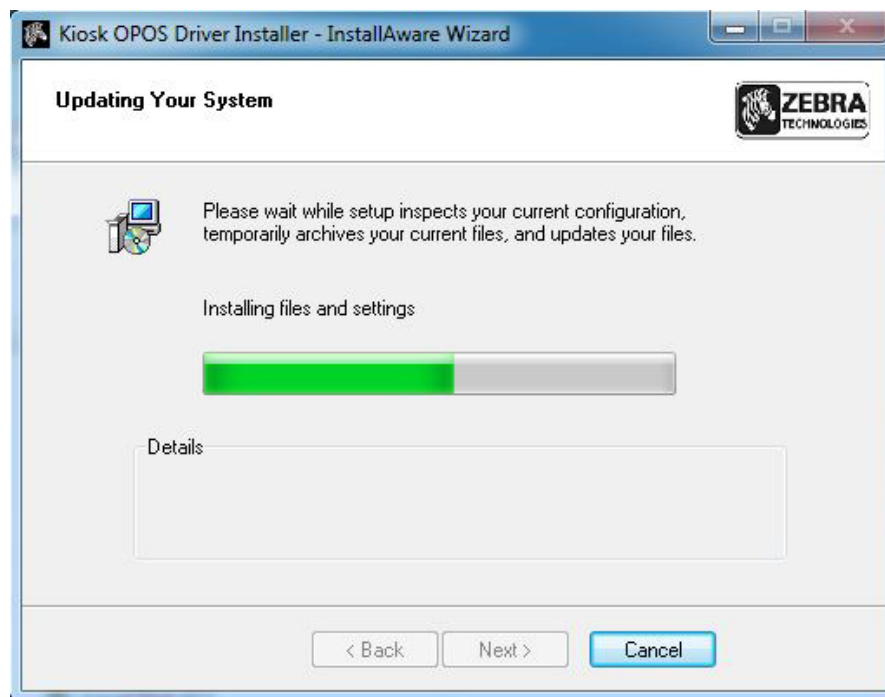
**10. Read the Important Information, and then click Next.**

A message appears indicating that the setup is completing.



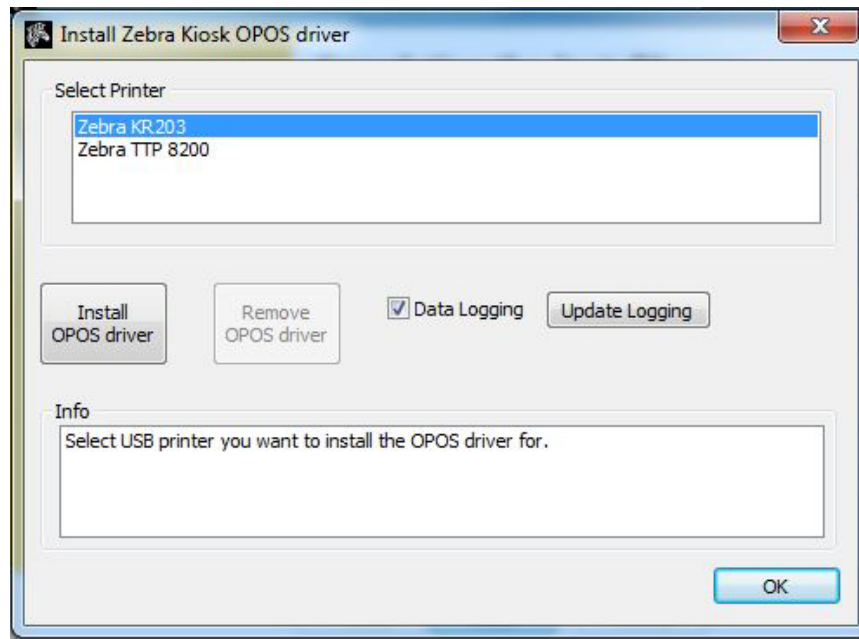
11. Click **Next** to begin the installation.

The Updating Your System dialog appears.



**Note** • A command prompt appears briefly as the files are being copied.

When the update is complete, the following dialog appears.



- 12.** In the **Select Printer** area, select the Kiosk printer that you want to use with the OPOS driver, click **Install OPOS driver**, and then click **OK**.

The following dialog appears indicating that the installer setup is complete.



- 13.** Click the **Run Kiosk OPOS Driver Installer now** check box, and then click **Finish**.

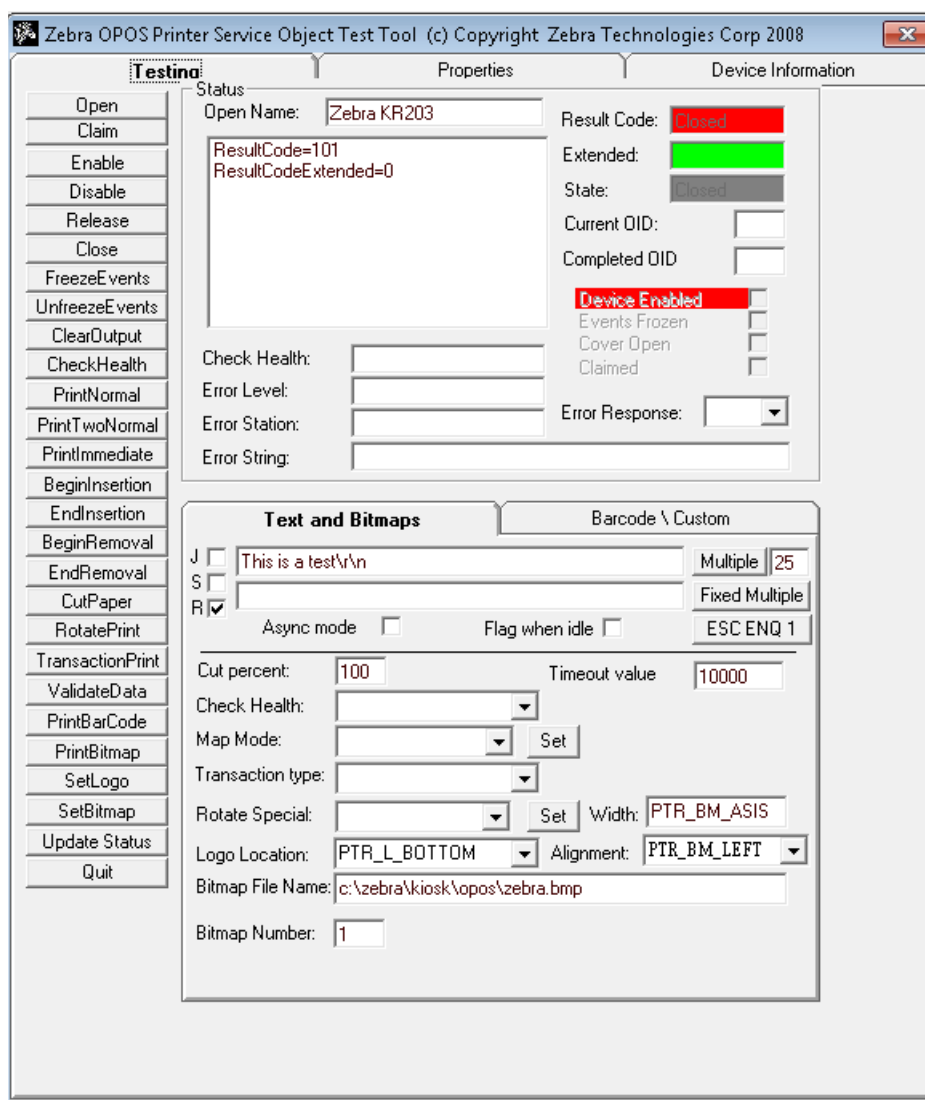
## Testing the Installation

When the driver is installed a link to the test application can be found in the Start menu OPOS\_Test folder or All Programs\Kiosk OPOS Driver Installer folder.

1. Click **Start > All Programs > Kiosk OPOS Driver Installer > OPOS\_Test**.



The Zebra OPOS Printer Service Object Test Tool appears.



2. In the **Open Name** box, type the full name of the Kiosk printer (e.g., Zebra TTP 2030, Zebra KR203).
3. Click **Open**, **Claim**, then **Enable** to connect the OPOS Printer Service Object Test Tool with the printer.
  - When connected, the Result Code, Extended, and State turn green.
  - If the printer is not in error state the Error Level, Error Station and Error String will also turn green.
  - Check Health will check the status and return successful or not successful, and will print a printer test page.
4. Exercise PrintNormal or other command buttons and at the end click CutPaper and a page will print.

The Error String will change and show the Paper in presenter.
5. Remove the paper and the status changes again.
6. To finish the test click **Disable**, **Release** and **Close** to disconnect the application from the printer.



# Kiosk Modifications of the OLE for Retail POS

## OLE for Retail POS Controls

The goal of this document is to provide an overview and programming guide for the Zebra Kiosk OPOS driver implementation.

For full reference of OLE for Retail POS please download the full manual at <http://www.nrf-arts.org/UnifiedPOS/UnifiedPOS%20Specification%20v1.11.zip>.

Due to the nature of the Zebra Kiosk printers not being true POS devices, we have certain limitations and differences to the original specification. The Zebra Kiosk OPOS driver is based on the Version 1.11 OLE for Retail POS specification but is working together with an underlying Windows driver for the Kiosk printers.

The following sections describe:

- *How an Application Uses an OPOS Control* [on page 16](#)
- *Device Power Reporting Model* [on page 16](#)
- *Summary of Common Properties, Methods, and Events* [on page 17](#)
- *DirectIO Method* [on page 19](#)
- *Events* [on page 21](#)
- *POS Printer Summary* [on page 21](#)
- *General Information* [on page 28](#)
- *Properties* [on page 33](#)
- *Methods* [on page 36](#)

## How an Application Uses an OPOS Control

The first action the application must take on the Control is to call its **Open** method. The parameter of this method selects a device name to associate with the Control. The **Open** method performs the following steps:

- Establishes a link to the device name that in our case is the Windows printer driver name.
- Initializes the properties **OpenResult**, **Claimed**, **DeviceEnabled**, **DataEventEnabled**, **FreezeEvents**, **AutoDisable**, **DataCount**, and **BinaryConversion**, as well as descriptions and version numbers of the OPOS Control layers. Additional class-specific properties may also be initialized.

Several applications may have an OPOS Control open at the same time. Therefore, after the device is opened, the application will need to call the **ClaimDevice** method to gain exclusive access to the device. The device must be claimed before the Control allows access to its methods and properties. Claiming the device ensures that other applications do not interfere with the use of the device. The application may call the **ReleaseDevice** method when the device can be shared by other applications – for instance, at the end of a transaction.

Before using the device, the application must set the **DeviceEnabled** property to TRUE. This value brings the device to an operational state, while FALSE disables the device.

After the application has finished using the device, the **DeviceEnabled** property should be set to FALSE, then the **ReleaseDevice** method and finally the **Close** method should be called to release the device and associated resources. Before exiting, an application should close all open OPOS Controls.

In summary, the application follows this general sequence:

- **Open** method: Call to link the Control Object to the Service Object.
- **ClaimDevice** method: Call to gain exclusive access to the device. Required for exclusive-use devices; optional for some sharable devices.
- **DeviceEnabled** property: Set to TRUE to make the device operational.
- Use the device.
- **DeviceEnabled** property: Set to FALSE to disable the device.
- **ReleaseDevice** method: Call to release exclusive access to the device.
- **Close** method: Call to release the Service Object from the Control Object.

## Device Power Reporting Model

Kiosk Printer OPOS segments device power into two states:

- **ONLINE**: The device is powered on and ready for use. This is the “operational” state.
- **OFF\_OFFLINE**: The device is either off or offline and the Service Object cannot distinguish these states.

Power reporting only occurs while the device is open, claimed (if the device is exclusive-use), and enabled.

## Summary of Common Properties, Methods, and Events

**Table 1 • Common Properties**

Name	Ver	Type	Access	Zebra Kiosk Printer Supported Properties
AutoDisable	1.2	Boolean	R/W	NO
CapCompareFirmwareVersion	1.9	Boolean	R	NO
BinaryConversion	1.2	Long	R/W	NO
CapPowerReporting	1.3	Long	R	NO
CapStatisticsReporting	1.8	Boolean	R	NO
CapUpdateFirmware	1.9	Boolean	R	NO
CapUpdateStatistics	1.8	Boolean	R	NO
CheckHealthText	1.0	String	R	YES
Claimed	1.0	Boolean	R	YES
DataCount	1.2	Long	R	NO
DataEventEnabled	1.0	Boolean	R/W	NO
DeviceEnabled	1.0	Boolean	R/W	YES
FreezeEvents	1.0	Boolean	R/W	YES
OpenResult	1.5	Long	R	YES
OutputID	1.0	Long	R	NO
PowerNotify	1.3	Long	R/W	NO
PowerState	1.3	Long	R	NO
ResultCode	1.0	Long	R	YES
ResultCodeExtended	1.0	Long	R	YES
State	1.0	Long	R	YES
ControlObjectDescription	1.0	String	R	YES
ControlObjectVersion	1.0	Long	R	YES
ServiceObjectDescription	1.0	String	R	YES
ServiceObjectVersion	1.0	Long	R	YES
DeviceDescription	1.0	String	R	YES
DeviceName	1.0	String	R	YES

**Table 2 • Common Methods**

<b>Name</b>	<b>Ver</b>	<b>Zebra Kiosk Printer Supported Methods</b>
Open	1.0	YES
Close	1.0	YES
ClaimDevice	1.0	YES
Claim	—	Special addition
ReleaseDevice	1.0	YES
CheckHealth	1.0	NO
ClearInput	1.0	NO
clearInputProperties	1.1	NO
ClearOutput	1.0	NO
DirectIO	1.0	YES
compareFirmwareVersion	1.9	NO
resetStatistics	1.8	NO
retrieveStatistics	1.8	NO
updateFirmware	1.9	NO
updateStatistics	1.8	NO

**Table 3 • Common Events**

<b>Name</b>	<b>Ver</b>	<b>Zebra Kiosk Printer Supported Events</b>
DataEvent	1.0	NO
DirectIOEvent	1.0	YES
ErrorEvent	1.0	YES
OutputCompleteEvent	1.0	NO
StatusUpdateEvent	1.0	YES

## DirectIO Method

**Syntax** `LONG DirectIO (LONG Command, LONG* pData, BSTR* pString);`

**Parameters** This table identifies the parameters for this format:

Parameter	Description
Command	Command number. 0 is the only valid option for the Kiosk printers and then you can send any valid ESC commands with the exception of status enquiries or data to the printer in the pString buffer. pData is set to the amount of bytes in pString.
pData	Pointer to additional numeric data. Set to the amount of bytes in pString.
pString	Pointer to additional string of valid ESC commands or data. The data format has to comply with the data format specified in the Kiosk printers Technical Manual. It is recommended to only use printer parameter set commands since the printing is implemented through the Kiosk Printer Windows Driver.

**Remarks** Call to communicate directly with the Service Object.

This method provides a means for a Service Object to provide functionality to the application that is not otherwise supported by the standard Control Object for its device class. Depending upon the Service Object's definition of the command, this method may be asynchronous or synchronous.

Use of **DirectIO** will make an application non-portable. The application may, however, maintain portability by performing **DirectIO** calls within conditional code. This code may be based upon the value of the **ServiceObjectDescription**, **DeviceDescription**, or **DeviceName** property.

**Return** One of the following values is returned by the method and placed in the **ResultCode** property:

Value	Meaning
OPOS_SUCCESS	Direct I/O successful.
Other Values	See <b>ResultCode</b> .

The use of DirectIO has one valid command entry.

`DirectIO (0, BufferLength, Buffer)`

The calling application has to prepare the DirectIO function with the following data: {STX} and the printout with the following data: {ETX}.

After sending of {STX} you can send all valid KPL commands to the printer but they have to be in binary format.

➔ **Example • ESC N 1** should be sent as 27 N 1. Where 27 is the binary decimal value for ESC.

➔ **Example • {STX}**

```
<ESC>N<1>Zebra Technologies<CR><LF>
<ESC>N<1>lincoln, RI, 02865<CR><LF>
<CR><LF>
<ESC>N<1>10/08/09                12:04<CR><LF>
<CR><LF>
<ESC>N<1>OOS Driver DirectIO sample<CR><LF>
<CR><LF>
<ESC>N<0>NO. : 000192<CR><LF>
<ESC>N<0>Date: 10/08/2009<CR><LF>
<ESC>N<0>Printouts: 1<CR><LF>
<ESC>N<0>OPOS driver for Kiosk use<CR><LF>
<ESC>N<0>Receipt printing<CR><LF>
<ESC>N<0>Theater tickets<CR><LF>
<ESC>N<0>etc.<CR><LF>
<ESC>N<0>No. : 000192<CR><LF>
<ESC>N<0>Zebra Technologies<CR><LF>
<CR><LF>
<CR><LF>
<ESC>N<1>Signature: _____<CR><LF>
<ESC>N<1>JOE DOE<CR><LF>
<CR><LF>
<ESC>BS<0><0><h 48><0><0><0><h 40><0><2><2>
<ESC>BW<0>733104000099<0>
<ESC>BS<1><0><h 28><0><0><0><h 40><4><2><2>
<ESC>BW<1><193>JOEDOE<0>
<RS>
{ETX}
```

Entering a command of 0 or 1 will result in printing the buffer with KPL and ending the page after the command. There is no continuation of the data possible with these Commands.

## Events

### ErrorEvent Event

**Remarks** Fired when an error is detected and the Control's **State** transitions into the error state.

Input error events are not delivered until the **DataEventEnabled** property is TRUE, so that proper application sequencing occurs.



**Note** • In the Kiosk environment it is not expected that the receipt will be retried because when a severe Error occurs, a Technician has to go out and clear the Error at the remote location. Therefore, Retry is not supported and Cancel will clear the printer buffer. All data will be sent to the printer regardless of the error state.

## POS Printer Summary

Table 4 • Common Properties

Property (Common)	Ver	Type	Access	May Use After	Zebra Kiosk Printer Supported Properties
AutoDisable	1.2	Boolean	R/W	Not Supported	NO
BinaryConversion	1.2	Long	R/W	Open	NO
CapCompareFirmwareVersion	1.9	Boolean	R	—	NO
CapPowerReporting	1.3	Long	R	Open	NO
CapStatisticsReporting	1.8	Boolean	R	Open	NO
CapUpdateFirmware	1.9	Boolean	R	Open	NO
CapUpdateStatistics	1.8	Boolean	R	Open	NO
CheckHealthText	1.0	String	R	Open	NO
Claimed	1.0	BooleanR	R	Open	YES
DataCount	1.2	Long	R	Not Supported	NO
DataEventEnabled	1.0	Boolean	R/W	<i>Not Supported</i>	NO
DeviceEnabled	1.0	Boolean	R/W	Open & Claim	YES
FreezeEvents	1.0	Boolean	R/W	Open	YES
OpenResult	1.5	Long	R	—	YES
OutputID	1.0	Long	R	Open	YES
PowerNotify	1.3	Long	R/W	Open	NO
PowerState	1.3	Long	R	Open	NO
ResultCode	1.0	Long	R	—	YES
ResultCodeExtended	1.0	Long	R	Open	YES
State	1.0	Long	R	—	YES

**Table 4 • Common Properties (Continued)**

<b>Property (Common)</b>	<b>Ver</b>	<b>Type</b>	<b>Access</b>	<b>May Use After</b>	<b>Zebra Kiosk Printer Supported Properties</b>
ControlObjectDescription	1.0	String	R	—	YES
ControlObjectVersion	1.0	Long	R	—	YES
ServiceObjectDescription	1.0	String	R	Open	YES
ServiceObjectVersion	1.0	Long	R	Open	YES
DeviceDescription	1.0	String	R	Open	YES
DeviceName	1.0	String	R	Open	YES

**Table 5 • Specific Properties**

<b>Property (Specific)</b>	<b>Ver</b>	<b>Type</b>	<b>Access</b>	<b>May Use After</b>	<b>Zebra Kiosk Printer Supported Properties</b>
CapCharacterSet	1.1	Long	R	Open	YES
CapConcurrentJrnRec	1.0	Boolean	R	Open	NO
CapConcurrentJrnSlp	1.0	Boolean	R	Open	NO
CapConcurrentRecSlp	1.0	Boolean	R	Open	NO
CapCoverSensor	1.0	Boolean	R	Open	YES
CapTransaction	1.1	Boolean	R	Open	NO
CapConcurrentPageMode	1.9	Boolean	R	Open	NO
CapMapCharacterSet	1.7	Boolean	R	Open	NO
CapJrnPresent	1.0	Boolean	R	Open	NO
CapJrn2Color	1.0	Boolean	R	Open	NO
CapJrnBold	1.0	Boolean	R	Open	NO
CapJrnCartridgeSensor	1.5	Long	R	Open	NO
CapJrnColor	1.5	Long	R	Open	NO
CapJrnDhigh	1.0	Boolean	R	Open	NO
CapJrnDwide	1.0	Boolean	R	Open	NO
CapJrnDwideDhigh	1.0	Boolean	R	Open	NO
CapJrnEmptySensor	1.0	Boolean	R	Open	NO
CapJrnItalic	1.0	Boolean	R	Open	NO
CapJrnNearEndSensor	1.0	Boolean	R	Open	NO
CapJrnUnderline	1.0	Boolean	R	Open	NO
CapRecPresent	1.0	Boolean	R	Open	YES
CapRec2Color	1.0	Boolean	R	Open	NO
CapRecBarCode	1.0	Boolean	R	Open	YES



Table 5 • Specific Properties (Continued)

Property (Specific)	Ver	Type	Access	May Use After	Zebra Kiosk Printer Supported Properties
CapRecBitmap	1.0	Boolean	R	Open	YES
CapRecBold	1.0	Boolean	R	Open	YES
CapRecCartridgeSensor	1.5	Long	R	Open	NO
CapRecColor	1.5	Long	R	Open	NO
CapRecDhigh	1.0	Boolean	R	Open	YES
CapRecDwide	1.0	Boolean	R	Open	YES
CapRecDwideDhigh	1.0	Boolean	R	Open	YES
CapRecEmptySensor	1.0	Boolean	R	Open	YES
CapRecItalic	1.0	Boolean	R	Open	YES
CapRecLeft90	1.0	Boolean	R	Open	YES
CapRecMarkFeed	1.5	Long	R	Open	NO
CapRecNearEndSensor	1.0	Boolean	R	Open	YES
CapRecPapercut	1.0	Boolean	R	Open	YES
CapRecRight90	1.0	Boolean	R	Open	NO
CapRecRotate180	1.0	Boolean	R	Open	NO
CapRecStamp	1.0	Boolean	R	Open	NO
CapRecUnderline	1.0	Boolean	R	Open	YES
CapRecPageMode	1.9	Boolean	R	Open	NO
CapSlpPageMode	1.9	Boolean	R	Open	NO
CapSlpPresent	1.0	Boolean	R	Open	NO
CapSlpFullslip	1.0	Boolean	R	Open	NO
CapSlp2Color	1.0	Boolean	R	Open	NO
CapSlpBarCode	1.0	Boolean	R	Open	NO
CapSlpBitmap	1.0	Boolean	R	Open	NO
CapSlpBold	1.0	Boolean	R	Open	NO
CapSlpBothSidesPrint	1.5	Boolean	R	Open	NO
CapSlpCartridgeSensor	1.5	Long	R	Open	NO
CapSlpColor	1.5	Long	R	Open	NO
CapSlpDhigh	1.0	Boolean	R	Open	NO
CapSlpDwide	1.0	Boolean	R	Open	NO
CapSlpDwideDhigh	1.0	Boolean	R	Open	NO
CapSlpEmptySensor	1.0	Boolean	R	Open	NO
CapSlpItalic	1.0	Boolean	R	Open	NO
CapSlpLeft90	1.0	Boolean	R	Open	NO

**Table 5 • Specific Properties (Continued)**

<b>Property (Specific)</b>	<b>Ver</b>	<b>Type</b>	<b>Access</b>	<b>May Use After</b>	<b>Zebra Kiosk Printer Supported Properties</b>
CapSlpNearEndSensor	1.0	Boolean	R	Open	NO
CapSlpRight90	1.0	Boolean	R	Open	NO
CapSlpRotate180	1.0	Boolean	R	Open	NO
CapSlpUnderline	1.0	Boolean	R	Open	NO
AsyncMode	1.0	Boolean	R/W	Open	YES
CartridgeNotify	1.5	Long	R/W	Open	NO
CharacterSet	1.0	Long	R/W	Open, Claim, & Enable	YES
CharacterSetList	1.0	String	R	Open	YES
CoverOpen	1.0	Boolean	R	Open, Claim, & Enable	YES
ErrorLevel	1.1	Long	R	Open	YES
ErrorStation	1.0	Long	R	Open	YES
ErrorString	1.1	String	R	Open	YES
FontTypefaceList	1.1	String	R	Open	YES
FlagWhenIdle	1.0	Boolean	R/W	Open	NO
MapCharacterSet	1.7	Boolean	R/W	Open	NO
MapMode	1.0	Long	R/W	Open	YES
PageModeArea	1.9	String	R/W	Open	NO
PageModeDescriptor	1.9	Int32	R/W	Open	NO
PageModeHorizontalPosition	1.9	Int32	R/W	Open	NO
PageModePrintArea	1.9	String	R/W	Open	NO
PageModePrintDirection	1.9	Int32	R/W	Open	NO
PageModeStation	1.9	Int32	R/W	Open	NO
PageModeVerticalPosition	1.9	Int32	R/W	Open	NO
RotateSpecial	1.1	Long	R/W	Open	YES
JrnLineChars	1.0	Long	R/W	Open, Claim, & Enable	NO
JrnLineCharsList	1.0	String	R	Open	NO
JrnLineHeight	1.0	Long	R/W	Open, Claim, & Enable	NO
JrnLineSpacing	1.0	Long	R/W	Open, Claim, & Enable	NO
JrnLineWidth	1.0	Long	R	Open, Claim, & Enable	NO

Table 5 • Specific Properties (Continued)

Property (Specific)	Ver	Type	Access	May Use After	Zebra Kiosk Printer Supported Properties
JrnLetterQuality	1.0	Boolean	R/W	Open, Claim, & Enable	NO
JrnEmpty	1.0	Boolean	R	Open, Claim, & Enable	NO
JrnNearEnd	1.0	Boolean	R	Open, Claim, & Enable	NO
JrnCartridgeState	1.5	Long	R	Open, Claim, & Enable	NO
JrnCurrentCartridge	1.5	Long	R/W	Open, Claim, & Enable	NO
RecLineChars	1.0	Long	R/W	Open, Claim, & Enable	YES
RecLineCharsList	1.0	String	R	Open	YES
RecLineHeight	1.0	Long	R/W	Open, Claim, & Enable	NO
RecLineSpacing	1.0	Long	R/W	Open, Claim, & Enable	YES
RecLineWidth	1.0	Long	R	Open, Claim, & Enable	YES
RecLetterQuality	1.0	Boolean	R/W	Open, Claim, & Enable	NO
RecEmpty	1.0	Boolean	R	Open, Claim, & Enable	YES
RecNearEnd	1.0	Boolean	R	Open, Claim, & Enable	YES
RecSidewaysMaxLines	1.0	Long	R	Open, Claim, & Enable	YES
RecSidewaysMaxChars	1.0	Long	R	Open, Claim, & Enable	YES
RecLinesToPaperCut	1.0	Long	R	Open, Claim, & Enable	YES
RecBarCodeRotationList	1.1	String	R	Open	YES
RecBitmapRotationList	1.7	String	R	Open	NO
RecCartridgeState	1.5	Long	R	Open, Claim, & Enable	NO
RecCurrentCartridge	1.5	Long	R/W	Open, Claim, & Enable	NO
SlpLineChars	1.0	Long	R/W	Open, Claim, & Enable	NO

**Table 5 • Specific Properties (Continued)**

<b>Property (Specific)</b>	<b>Ver</b>	<b>Type</b>	<b>Access</b>	<b>May Use After</b>	<b>Zebra Kiosk Printer Supported Properties</b>
SlpLineCharsList	1.0	String	R	Open	NO
SlpLineHeight	1.0	Long	R/W	Open, Claim, & Enable	NO
SlpLineSpacing	1.0	Long	R/W	Open, Claim, & Enable	NO
SlpLineWidth	1.0	Long	R	Open, Claim, & Enable	NO
SlpLetterQuality	1.0	Boolean	R/W	Open, Claim, & Enable	NO
SlpEmpty	1.0	Boolean	R	Open, Claim, & Enable	NO
SlpNearEnd	1.0	Boolean	R	Open, Claim, & Enable	NO
SlpSidewaysMaxLines	1.0	Long	R	Open, Claim, & Enable	NO
SlpSidewaysMaxChars	1.0	Long	R	Open, Claim, & Enable	NO
SlpMaxLines	1.0	Long	R	Open, Claim, & Enable	NO
SlpLinesNearEndToEnd	1.0	Long	R	Open, Claim, & Enable	NO
SlpBarCodeRotationList	1.1	String	R	Open	NO
SlpBitmapRotationList	1.7	String	R	Open	NO
SlpPrintSide	1.5	Long	R	Open, Claim, & Enable	NO
SlpCartridgeState	1.5	Long	R	Open, Claim, & Enable	NO
SlpCurrentCartridge	1.5	Long	R/W	Open, Claim, & Enable	NO

**Table 6 • Common Methods**

<b>Method (Common)</b>	<b>Ver</b>	<b>May Use After</b>	<b>Zebra Kiosk Printer Supported Methods</b>
Open	1.0	—	YES
Close	1.0	Open	YES
ClaimDevice	1.0	Open	YES
ReleaseDevice	1.0	Open & Claim	YES

Table 6 • Common Methods

Method (Common)	Ver	May Use After	Zebra Kiosk Printer Supported Methods
CheckHealth	1.0	Open, Claim, & Enable	YES
ClearInput	1.0	Not Supported	NO
ClearOutput	1.0	Not Supported	NO
DirectIO	1.0	Open	YES
compareFirmwareVersion	1.9	—	NO
resetStatistics	1.8	—	NO
retrieveStatistics	1.8	—	NO
updateFirmware	1.9	—	NO
updateStatistics	1.8	—	NO

Table 7 • Specific Methods

Method (Specific)	Ver	May Use After	Zebra Kiosk Printer Supported Methods
PrintNormal	1.0	Open, Claim, & Enable	YES
PrintTwoNormal	1.0	Open, Claim, & Enable	NO
PrintImmediate	1.0	Open, Claim, & Enable	YES
printMemoryBitmap	1.10	—	NO
BeginInsertion	1.0	Open, Claim, & Enable	NO
EndInsertion	1.0	Open, Claim, & Enable	NO
BeginRemoval	1.0	Open, Claim, & Enable	NO
clearPrintArea	1.9	—	NO
EndRemoval	1.0	Open, Claim, & Enable	NO
CutPaper	1.0	Open, Claim, & Enable	Full cut and 50% cut to implement the page hold in the driver. The driver cut value will be used to partial cut.
RotatePrint	1.0	Open, Claim, & Enable	Only Landscape mode
PrintBarCode	1.0	Open, Claim, & Enable	YES
PrintBitmap	1.0	Open, Claim, & Enable	YES
TransactionPrint	1.1	Open, Claim, & Enable	NO
ValidateData	1.1	Open, Claim, & Enable	NO
SetBitmap	1.0	Open, Claim, & Enable	When deleting bitmaps all bitmaps will be deleted
SetLogo	1.0	Open, Claim, & Enable	NO

**Table 7 • Specific Methods**

Method (Specific)	Ver	May Use After	Zebra Kiosk Printer Supported Methods
ChangePrintSide	1.5	Open, Claim, & Enable	NO
MarkFeed	1.5	Open, Claim, & Enable	NO
pageModePrint	1.9	—	NO

**Table 8 • Events**

Name	Ver	May Occur After	Zebra Kiosk Printer Supported Events
DataEvent	1.0	Not Supported	NO
DirectIOEvent	1.0	Open, Claim	YES
ErrorEvent	1.0	Open, Claim, & Enable	YES
OutputCompleteEvent	1.0	Open, Claim, & Enable	NO
StatusUpdateEvent	1.0	Open, Claim, & Enable	YES

## General Information

The POS Printer follows the general output model, with some enhancements:

Printer out-of-paper and cover open conditions are reported by setting the **ResultCode** to OPOS\_E\_EXTENDED and then setting **ResultCodeExtended** to one of the following error conditions:

- OPOS\_EPTR\_REC\_EMPTY
- OPOS\_EPTR\_COVER\_OPEN
- OPOS\_EPTR\_REC\_CARTRIDGE\_REMOVED
- OPOS\_EPTR\_REC\_CARTRIDGE\_EMPTY
- OPOS\_EPTR\_REC\_HEAD\_CLEANING

Other printer errors are reported by setting the **ResultCode** to OPOS\_E\_FAILURE or another standard error status. These failures are typically due to a printer fault or jam, or to a more serious error. The Kiosk printer returns a few different error strings back which state the kind of failure.

## Data Characters and Escape Sequences

The following escape sequences are recognized. If an escape sequence specifies an operation that is not supported by the printer station, then it is ignored.

**One Shots** Perform indicated action.

**Table 9 • One Shot Data Characters and Escape Sequences**

Name	Data	Remarks	Zebra Kiosk Printer Supported
Paper cut	ESC  #P	Cuts receipt paper. The character '#' is replaced by an ASCII decimal string telling the percentage cut desired. If '#' is omitted, then a full cut is performed. For example: The C string "\x1B 75P" requests a 75% partial cut.	For a full cut use ESC 100P. Use ESC 50P if you want to use Hold Page in the Windows driver.
Feed and Paper cut	ESC  #fP	Cuts receipt paper, after feeding the paper by the <b>RecLinesToPaperCut</b> lines. The character '#' is defined by the "Paper cut" escape sequence.	See Paper cut. Use ESC 100P. This command is not working as expected when used with the Windows printer driver and the paper advance at the end before cut has to be setup in the Driver's Printing Preferences under Advanced settings.
Feed, Paper cut, and Stamp	ESC  #sP	Cuts and stamps receipt paper, after feeding the paper by the <b>RecLinesToPaperCut</b> lines. The character '#' is defined by the "Paper cut" escape sequence.	NO
Fire stamp	ESC  sL	Fires the stamp solenoid, which usually contains a graphical store emblem.	NO
Print bitmap	ESC  #B	Prints the pre-stored bitmap. The character '#' is replaced by the bitmap number. See setBitmap method.	YES <b>Note</b> • This command will be ignored in Left90 rotation.
Print top logo	ESC  tL	Prints the pre-stored top logo.	NO
Print bottom logo	ESC  bL	Prints the pre-stored bottom logo.	NO
Feed lines	ESC  #fF	Feed the paper forward by lines. The character '#' is replaced by an ASCII decimal string telling the number of lines to be fed. If '#' is omitted, then one line is fed.	YES

**Table 9 • One Shot Data Characters and Escape Sequences (Continued)**

Name	Data	Remarks	Zebra Kiosk Printer Supported
Feed units	ESC  #uF	Feed the paper forward by mapping mode units. The character '#' is replaced by an ASCII decimal string telling the number of units to be fed. If '#' is omitted, then one unit is fed.	NO
Feed reverse	ESC  #rF	Feed the paper backward. The character '#' is replaced by an ASCII decimal string telling the number of lines to be fed. If '#' is omitted, then one line is fed.	NO

**Print Mode** Characteristics that are remembered until explicitly changed.

**Table 10 • Print Mode Data Characters and Escape Sequences**

Name	Data	Remarks	Zebra Kiosk Printer Supported
Pass through embedded data	ESC  #E	Send the following # characters of data through to the hardware without modifying it. The character '#' is replaced by an ASCII decimal string telling the number of bytes following the escape sequence that should be passed through as-is to the hardware.	NO
Print in-line barcode	ESC  #R	Prints the defined barcode in-line. The character '#' is the number of characters following the R to use in the definition of the characteristics of the barcode to be printed. See details below.	NO
Font typeface selection	ESC  #fT	Selects a new typeface for the following data. Values for the character '#' are: 0 = Default typeface. 1 = Select second typeface from the <b>FontTypefaceList</b> property. 2 = Select third typeface from the <b>FontTypefaceList</b> property. And so on.	YES The Kiosk OPOS driver has 2 fonts implemented. Valid selections are 0 for (small) Monospace, 1 for Lucida Console (big)



**Print Line** Characteristics that are reset at the end of each print method or by a “Normal” sequence.

**Table 11 • Print Line Data Characters and Escape Sequences**

Name	Data	Remarks	Zebra Kiosk Printer Supported
Bold	ESC  (!)bC	Prints in bold or double-strike. If ‘!’ is specified then bold is disabled,	YES
Underline	ESC  #uC	Prints with underline. The character ‘#’ is replaced by an ASCII decimal string telling the thickness of the underline in printer dot units. If ‘#’ is omitted, then a printer-specific default thickness is used.	YES
Italic	ESC  (!)iC	Prints in italics. If ‘!’ is specified then italic is disabled.	YES
Alternate color (Custom)	ESC  #rC	Prints using an alternate custom color. The character ‘#’ is replaced by an ASCII decimal string indicating the desired color. The value of the decimal string is equal to the value of the cartridge constant used in the printer device properties. If ‘#’ is omitted, then the secondary color (Custom Color 1) is selected. Custom Color 1 is usually red.	NO
Reverse video	ESC  (!)rvC	Prints in a reverse video format. If ‘!’ is specified then reverse video is disabled.	YES
Shading	ESC  #sC	Prints in a shaded manner. The character ‘#’ is replaced by an ASCII decimal string telling the percentage shading desired. If ‘#’ is omitted, then a printer-specific default level of shading is used.	NO
Single high & wide	ESC  1C	Prints normal size.	YES
Double wide	ESC  2C	Prints double-wide characters.	YES, can co-exist with normal characters on the line
Double high	ESC  3C	Prints double-high characters.	YES, can co-exist with normal characters on the line
Double high & wide	ESC  4C	Prints double-high/double-wide characters.	YES, can co-exist with normal characters on the line

**Table 11 • Print Line Data Characters and Escape Sequences (Continued)**

<b>Name</b>	<b>Data</b>	<b>Remarks</b>	<b>Zebra Kiosk Printer Supported</b>
Scale horizontally	ESC  #hC	Prints with the width scaled ‘#’ times the normal size, where ‘#’ is replaced by an ASCII decimal string.	YES, max value 7
Scale vertically	ESC  #vC	Prints with the height scaled ‘#’ times the normal size, where ‘#’ is replaced by an ASCII decimal string.	YES, max value 15
RGB color	ESC  #fC	Prints in # color. The character ‘#’ is replaced by an ASCII decimal string indicating the additive amount of RGB to produce the desired color. There are 3 digits each of Red, Green and Blue elements. Valid values range from “000” to “255”. (For example, “255255000” represents yellow.) Color Matching to the subtractive percentage of CMY (Cyan, Magenta and Yellow components) to produce the desired color matching specified by the RGB is up to the service object. If ‘#’ is omitted, then the primary color is used. Bitmap printing is not affected.	NO
Center	ESC  cA	Aligns following text in the center.	YES
Right justify	ESC  rA	Aligns following text at the right.	YES
Normal	ESC  N	Restores printer characteristics to normal condition.	YES
SubScript	ESC  (!)tbC	Prints SubScript characters. If ‘!’ is specified then SubScript is disabled.	NO
SuperScript	ESC  (!)tpC	Prints SuperScript characters. If ‘!’ is specified then SuperScript is disabled.	NO

## Properties

### CapCharacterSet Property

*Added in Release 1.1, Updated in Release 1.5*

**Syntax** LONG CapCharacterSet

**Remarks** Holds the default character set capability. It is PTR\_CCS\_WINDOWS for Kiosk printer.

Value	Meaning
PTR_CCS_WINDOWS	The Windows ANSI character set. The value of this constant is 999. This is exactly equivalent to the Windows code page 1252.

This property is initialized by the **Open** method.

### CapRecMarkFeed Property

*Added in Release 1.5*

Mark feed is only supported through settings in the Windows driver.

### CapRecNearEndSensor Property

- This property is initialized by the Open method.
- This sensor will be reported as “Paper near end.”

### CapRecPapercut Property

- This property is initialized by the Open method.
- The Kiosk printer supports full 100% cut and 50% cut. See CutPaper on [page 27](#) for more information.

### CharacterSet Property

*Updated in Release 1.5*

This property is initialized when the device is first enabled following the Open method.

Values are: PTR\_CS\_WINDOWS for Kiosk printer.

Value	Meaning
PTR_CS_WINDOWS	The Windows ANSI character set. The value of this constant is 999. This is exactly equivalent to the Windows code page 1252.

## CharacterSetList Property

“999” for Kiosk printer.

## FontTypefaceList Property

### *Added in Release 1.1*

There are two fonts supported in the Kiosk printer; a small Monospac821 BT and a larger Lucida Console.

## MapMode Property

The value of MapMode is initialized to PTR\_MM\_DOTS when the device is first enabled following the Open method. This is the only valid setting for the Kiosk printer.

## RecBarCodeRotationList Property

### *Added in Release 1.1*

The Kiosk printer cannot rotate barcodes. The barcode is always in Portrait position.

## RecLineChars Property

The value of RecLineChars is initialized to the printer’s default line character width when the device is first enabled following the Open method. It depends on the page width setting of the Paper setting in the Windows driver and can be controlled by modifying the paper template used in the driver.

## RecLineCharsList Property

This property is initialized by the Open method. The string consists of ASCII numeric set numbers, separated by commas.

It depends on the page width setting of the Paper setting in the Windows driver and can be controlled by modifying the paper template used in the driver.

## RecLineHeight Property

The value of RecLineHeight is initialized to the printer’s default line height when the device is first enabled following the Open method. It depends on the selected Windows font height and is 30 for the first font and 44 for the second font. The height is reflected in logical units.

## RecLineSpacing Property

The value of RecLineSpacing is initialized to the printer's default line spacing when the device is first enabled following the Open method. The value is approximately 7/10 of the RecLineHeight.

## RecLinesToPaperCut Property

This property is initialized when the device is first enabled following the Open method. The value is determined by the CutPosition parameter set during installation (found in the INF file and set in mm) and divided by the RecLineHeight.

## RecLineWidth Property

The value of RecLineWidth is initialized to the printer's default line width when the device is first enabled following the Open method.

It depends on the selected Windows font height and is 15 for the first font and 22 for the second font.

## RecSidewaysMaxChars Property

This property is initialized when the device is first enabled following the Open method. The number depends on the paper template used in the Windows driver and the printing mode the driver is set to.

## RecSidewaysMaxLines Property

This property is initialized when the device is first enabled following the Open method.

The number depends on the paper template used in the Windows driver and the printing mode the driver is set to.

## RotateSpecial Property

### ***Added in Release 1.1***

This property is initialized to PTR\_RP\_NORMAL by the Open method.

Values are: Only PTR\_RP\_NORMAL is valid for the Kiosk printer.

## Methods

### CutPaper Method

The Percentage parameter indicates the percentage of paper to cut. The constant identifier PTR\_CP\_FULLLCUT or the value 100 causes a full paper cut. A full cut and 50% cut are the only valid values for the Kiosk printer.

### PrintBarcode Method

	Description
Symbology	Barcode symbol type to use. See values below.
Height	Barcode height. Expressed in the unit of measure given by MapMode. (Valid values can be 96, 124.)
Width	Barcode width. Expressed in the unit of measure given by MapMode. (Valid values can be 40 to 70.)
Alignment	Placement of the barcode. The Kiosk printer can't align Barcode other then PTR_BC_LEFT.
TextPosition	Placement of the readable character string. See values below.

The Alignment parameter values are:

Value	Meaning
PTR_BC_LEFT	Align with the left-most print column.
PTR_BC_CENTER	Align in the center of the station. (Not valid.)
PTR_BC_RIGHT	Align with the right-most print column. (Not valid.)
Other Values	Distance from the left-most print column to the start of the barcode. Expressed in the unit of measure given by MapMode. (Not valid.)

The TextPosition parameter values are:

Value	Meaning
PTR_BC_TEXT_NONE	No text is printed. Only print the barcode.
PTR_BC_TEXT_ABOVE	Print the text above the barcode.
PTR_BC_TEXT_BELOW	Print the text below the barcode.

The Symbology parameter values for this release are:

**Table 12 • One Dimensional Symbologies**

Value	Meaning	
PTR_BCS_UPCA	UPC-A	—
PTR_BCS_UPCA_S	UPC-A with supplemental barcode	(Not valid)
PTR_BCS_UPCE	UPC-E	(Not valid)
PTR_BCS_UPCE_S	UPC-E with supplemental barcode	(Not valid)
PTR_BCS_UPCD1	UPC-D1	(Not valid)
PTR_BCS_UPCD2	UPC-D2	(Not valid)
PTR_BCS_UPCD3	UPC-D3	(Not valid)
PTR_BCS_UPCD4	UPC-D4	(Not valid)
PTR_BCS_UPCD5	UPC-D5	(Not valid)
PTR_BCS_EAN8	EAN 8 (= JAN 8)	(Not valid)
PTR_BCS_JAN8	JAN 8 (= EAN 8)	(Not valid)
PTR_BCS_EAN8_S	EAN 8 with supplemental barcode	(Not valid)
PTR_BCS_EAN13	EAN 13 (= JAN 13)	(Only EAN 13)
PTR_BCS_JAN13	JAN 13 (= EAN 13)	(Not valid)
PTR_BCS_EAN13_S	EAN 13 with supplemental barcode	(Not valid)
PTR_BCS_EAN128	EAN-128	(Not valid)
PTR_BCS_TF	Standard (or discrete) 2 of 5	(Not valid)
PTR_BCS_ITF	Interleaved 2 of 5	(Not valid)
PTR_BCS_Codabar	Codabar	(Not valid)
PTR_BCS_Code39	Code 39	(Not valid)
PTR_BCS_Code93	Code 93	(Not valid)
PTR_BCS_Code128	Code 128	(Not valid)
PTR_BCS_OCRA	OCR “A”	(Not valid)
PTR_BCS_OCRB	OCR “B”	(Not valid)

**Table 13 • Two Dimensional Symbologies**

Value	Meaning	
PTR_BCS_PDF417	PDF 417	(Not valid)
PTR_BCS_MAXICODE	MAXICODE	(Not valid)

**Remarks** Call to print a barcode on the specified printer station.

If the property RotateSpecial indicates that the barcode is to be rotated, then perform the rotation. The Height, Width, and TextPosition parameters are applied to the barcode before the rotation. For example, if PTR\_BC\_TEXT\_BELOW is specified and the barcode is rotated left, then the text will appear on the paper to the right of the barcode.



**Note** • Rotation is not possible.

## PrintBitmap Method

Parameter	Description
Station	The printer station to be used. Only PTR_S_RECEIPT can be used.
FileName	Name of Windows bitmap file. The file must be in uncompressed and 2 color (black and white) format. Color inversion may be necessary.
Width	Printed width of the bitmap to be performed. Only PTR_BM_ASIS can be used.
Alignment	Placement of the bitmap. (No alignment other than left-most print column possible (PTR_BM_LEFT).)



**Note** • PTR\_BM\_ASIS is the only valid Width value for the Kiosk printer.

## RotatePrint Method

**Rotation** Direction of rotation. See values below.



Value	Meaning
PTR_RP_RIGHT90	Rotate printing 90° to the right (clockwise). (Not valid.)
PTR_RP_LEFT90	Rotate printing 90° to the left (counter-clockwise). The printer prints in Landscape mode and can't print bitmaps within OPOS commands. The printer will not print any text until it is switched back into Normal, the page is full, or receives a FF to print and eject the page. You need to set the RecLineChars to define the page width.
PTR_RP_ROTATE180	Rotate printing 180°, that is, print upside-down. (Not valid.)
PTR_RP_NORMAL	End rotated printing.



**Note** • The driver will send an FF to start the print out and eject the page after printing.



## SetBitmap Method

Parameter	Description
BitmapNumber	<p>The number to be assigned to this bitmap. Two bitmaps, numbered 1 and 2, may be set. You have to set Bitmap 1 first and then Bitmap 2 or you receive an Error.</p> <p> <b>Note</b> • You cannot overwrite a bitmap. In order to reset the bitmap you have to unset all bitmaps first and then set the new bitmaps again. To do this you have to SetBitmap with an empty string.</p>
FileName	<p>Name of Windows bitmap file. The file must be in uncompressed and 2 color (black and white) format. Color inversion may be necessary.</p> <p>If set to an empty string (“”), then the bitmap is unset.</p> <p> <b>Note</b> • When the bitmaps are unset, all bitmaps are lost.</p>

**Remarks** Call to save information about a bitmap for later printing.

The bitmaps may be no bigger than approximately 5000 bytes all together otherwise the printer cannot load it.

Only 2 bitmaps may be set, and each bitmap number may only be used for one station at a time.







**Zebra Technologies Corporation**

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, 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>