Transition Guide –

For customers moving from Legacy (P1XXi, PX330i/430i) printers, to ZXP 3 or ZXP 7 printers.

The purpose of this guide is to provide basic information and guidance for customers that used Eltron Printer Card Language (EPCL) commands to print text, Barcodes, and or Magnetically Encoded cards with legacy printers (e.g. P1xx and P3xx printers) and want to transition to ZXP Card printers.

Zebra has several options for integrating ZXP Card printers into new or legacy customer applications.

Integration Method	Advantages	Requirements or Pre-Requisites	Typical Use cases
Use Zebra Card Printer SDK	Allows the application and application developer complete printer control and print and encode functionality. Supports all ZXP printer features and functions.	Requires host level programing skills to integrate print functionality into application. Different Zebra Card SDK's may be required depending upon Windows, or Java Environment and card printer models.	Applications that need full print and encode features. System integrators that want to tightly integrate Card Printing into their application or workflow. Examples include Full color printing, Technology card deployments, and all ribbon or media types.
Template Printing from Zebra Virtual PrintWare – PrintManager	Application sends variable card data directly from application to PrintManager middleware for printing. Print from any Network based legacy application (including mobile devices, etc...) Minimal to no SDK programing, and allows fast application integration.	Print to Ethernet attached printers only. Need a windows machine to host PrintManager in application architecture	Event cards, (cards printed from mobile devices) Printing from an AS400 or other legacy application with limited SDK or driver support. Managed print jobs to distributed network attached printers. Large scalable printer deployments
Printing Directly from Legacy application through the Zebra windows printer driver.	Minimal application change assuming the application can print to the print driver.	Effort is required to integrate the print job from legacy application workflow and print interfaces. Process requires the use of a Zebra Windows Driver or a Zebra Linux driver (Linux driver for ZXP 3.)	Large Enterprise deployments where Card issuance is coupled with logic or workflow tied to business processes such as printing from SAP or Oracle based applications.

Frequently Asked Questions -

Q: Why did Zebra change from the EPCL printer language to the current processes?
A: The ZXP printers represent significant improvements in throughput and print quality which could only be achieved by leveraging host PC processing power and updated electronics available through the ZRaster print process in the ZXP 1 and 3 printers and ZMotif Printer SDK for the ZXP 7 printer.

Q: If I'm only sending monochrome data to my current printer, will there be a limitation in what commands I can send to the ZXP printers?
A: You cannot send print commands directly to a ZXP printer. Any print commands must be sent to the printer through our driver or SDK.

Q: Do I have to migrate to network based printers?
A: No. Zebra Virtual Printware requires network printers but the ZXP Windows drivers, Zebra Linux drivers, or SDK allow deployment via USB or Network based printers.

Q: How do I download the SDK?
A: The SDK's are available through Zebra.com, Card Software, and Developers Tools. Click on the following link to access the developer tools and SDK information: https://www.zebra.com/us/en/products/software/card-printers.html#mainpartabscontainer_509b=developer-tools

Q: Where do I obtain a Linux driver for the ZXP 1 and 3?
A: The driver can be obtained by submitting a request found on the ZXP S1 or S1 Driver download page or by selecting this link: https://www.zebra.com/us/en/forms/linux-driver-zxps1s3-request.html

Q: How do I get Technical support for the SDK?
A: Send an E-mail to: Card_SDK_Questions@zebra.com.

Q: Where do I get more information on Template printing?
A: Additional information on Template printing using Zebra Virtual PrintWare PrintManager can be found at: https://www.zebra.com/us/en/products/software/developer-tools/zebra-virtual-printware.html

Q: Do you have any further documentation on integration alternatives?
A: Yes. Documentation and sample files can be found on the Zebra Knowledge Base at Zebra.com. The following link can direct you to the Zebra Knowledge Base: https://km.zebra.com/kb

Q: What is Application Level Programming for Card printer integration?
A: Adding functionality into an application where print functions are directly included into the application is done via our SDK and programming calls. The calls are compiled directly into an application which will require programming skills and access to an applications source code.

Q: What is Template Printing?

A: Template printing requires the addition of Zebra Virtual PrintWare PrintManager as a middleware application running on the network. Variable card data is sent to Print Queues within PrintManager, representing physical, card printers on the network. PrintManager then merges card variable data with the overall card design and sends the print job to the printer. This process separates the actual card printing process from the requesting application allowing for very easy card integration without significant development effort.

Q: How exactly does Template printing work?

A: Template Printing enables Card printing from a legacy application by separating the variable card print data from the overall card design and print issuance process. By using PrintManager template printing, you can integrate card printing into a network based application by sending the variable data as a text file to PrintManager to print a card. This allows for integration of card printing from non-windows legacy and mobile applications. Note: template printing requires the building of templates which represent those features of a card that are common to the entire card: company logo, company name, background images, etc... These templates must be stored on the PC hosting Print Manager prior to sending variable data as a text file to PrintManager. In addition to PrintManager, Card Studio is required to create the templates and to store them on the PC hosting PrintManager.

Q: What would template variable data sent from an application job look like?

A: The following is an example of an XML file containing the variable data to be sent from an AS400 or SAP system to PrintManager to generate a card print job.

CSV data document would look like:

```
template,Firstname,Lastname
temp1,John,Doe
```

XML data document would look like:

```
<data template=”text_job”>
        <Firstname>John</Firstname>
        <Lastname>Doe</Lastname>
</data>
```

Q: How large could my network be and still work with Template printing?

A: Contact your Zebra representative about Zebra Virtual PrintWare Template printing. There are multiple considerations in the overall planning and implementation of a successful template printing deployment.

Q: What are the benefits or advantages of using the Zebra Virtual PrintWare solution with template printing?

A: Card printing can be added easily, to any legacy application though a simple text based file with the variable card data. The Zebra Virtual PrintWare PrintManager manages all the complexity of merging the variable data with a specified card template design. Then, PrintManager automatically creates and manages the print job sent to the specified network Card printer.

Q: Can a print job for Legacy Zebra card printers be integrated into the SAP workflow and print interfaces for ZXP card printers?

A: Yes. Consult with an application engineer to ensure the SAP environment and workflow are understood. Knowing environment requirements for any application will allow for the fastest integration.

Appendices have been added at the end of this document to help explain the print process' for Legacy printers (P1xx, P3xx, P4xx), as well as ZXP 1, ZXP 3, and ZXP 7 card printers.

Appendix A

Print Process for Legacy printers.

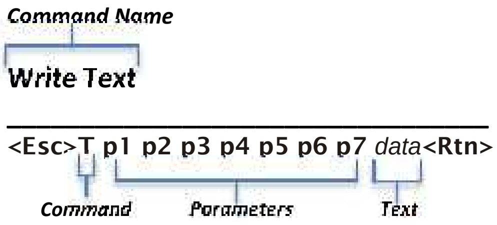
Legacy Print Process:

All printers employ a Common Printer Command Set plus additional Command Sets for any model-specific features (e.g. Flip-over module for dual side printing, Magnetic encoder, Smart Card encoder, etc...).

Historically, these commands can be sent directly to the printer. Using this method associated programming for Printer application integration is not necessary. <=

Each command begins with a Command Initiator (the Escape Character). The Escape commands allow for printer set ups, many of which determine how a Printer Module Element reacts upon receiving an incoming card. For example, a data download must occur before the Printer Module can print a card or encode a magnetic stripe.

Typical command structure:



<Esc>T p1 p2 p3 p4 p5 p6 p7 *data***<Rtn>**

ASCII Programming Code for the same command

<Esc>T 100 100 0 1 20 30 1 Text<Rtn> ASCII Programming Equivalent

Hexadecimal Programming Code for the same command

1B5420313030203130302030203120323020333020312054657874 0D Hexadecimal Programing Equivalent

Space (Delimiters)

Escape (Command Initiator)

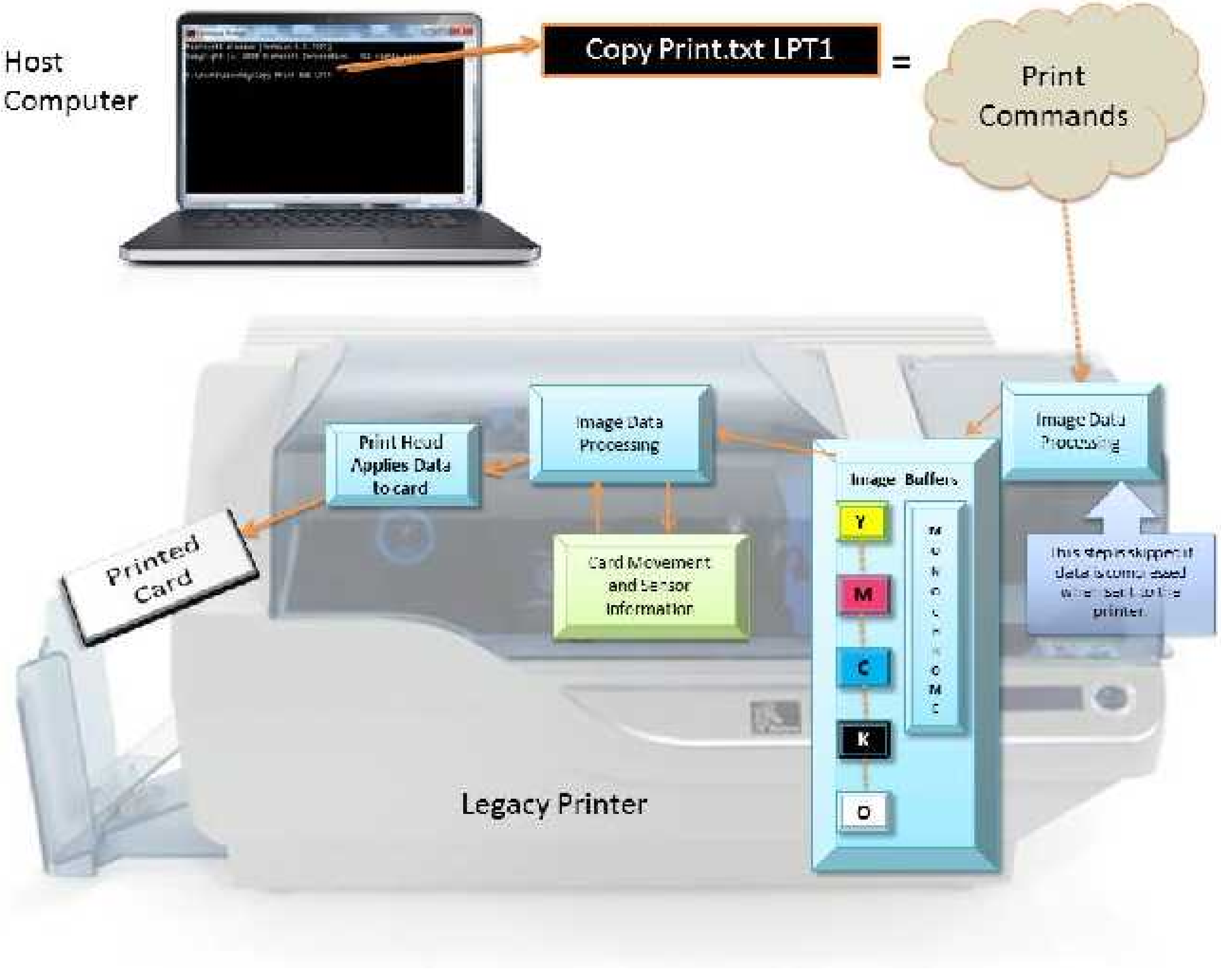
Carriage Return (Command Terminator)

Typically customers used an ASCII based Text Editors to create simple command files. In the DOS environment, MS-DOS EDIT was another choice. To execute the files, customers used the Print Command from the editor, or from DOS, the COPY Command, to send the file directly to the printer.

Examples using the COPY Command are:

```
COPY  file  name.ext  LPT1
```

Files could be sent to the printer compressed or uncompressed. If sent uncompressed, the printer would use an image data compression algorithm to compress the data and pass it to individual Image buffers. Legacy printers such as the P1xx or P3xx printers, contained image buffers for each of the ribbon color panels, Yellow, Magenta, Cyan, Kresin, and Overlay. Printers also contained a Monochrome image buffer for print jobs that involved monochrome ribbons. From the image buffers, data for each color panel or monochrome ribbon, was decompressed then sent individually, to the print head, for the proper color panel, starting with Yellow, then Magenta, etc….



Appendix B

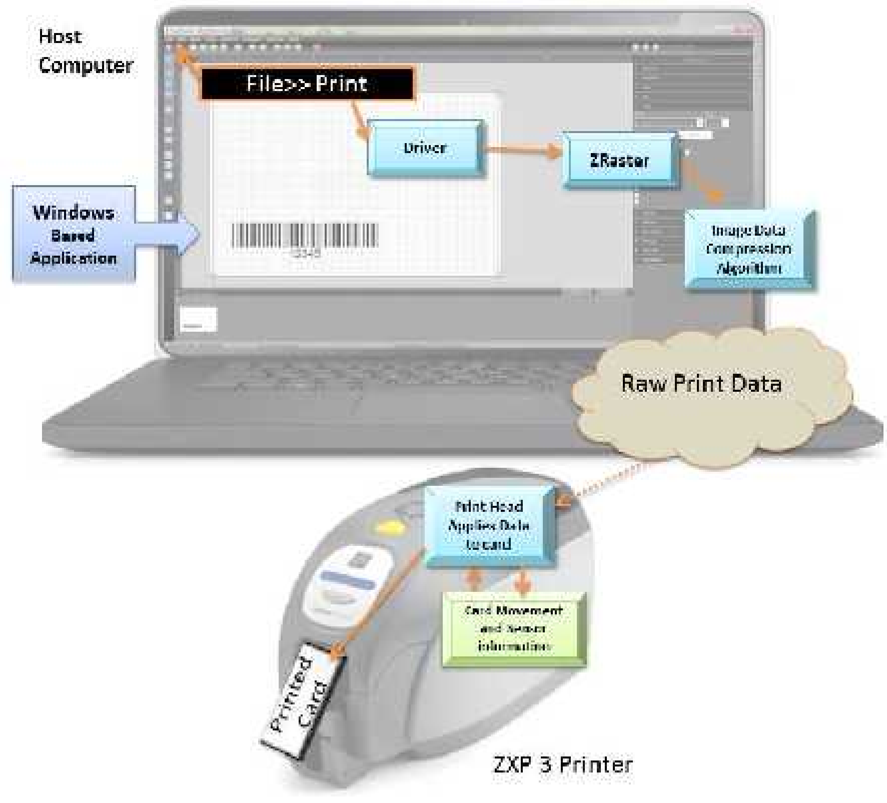
Print Process for ZXP 1 and ZXP 3 printers.

ZXP 1 and ZXP 3 Print Process:

ZXP 3 printers utilize a host based system called ZRaster in which the image processing of data is completed within the Host Computer and not the printer (prior to being set to the printer). Instead of rasterizing image print data directly in the printer firmware as Legacy printers do, ZXP 3 printers utilize the ZRaster algorithm to pre-process all print data in the host computer. The rasterized image data is sent to the printer as part of the print job. The Benefits of this include significantly faster print speed and better print quality.

Whether sending a print job via USB or Ethernet if sending a print job through Windows, utilizing the MS Windows Printer Driver simplifies the process. Whether you are sending the print job from a Windows application or the Windows based SDK the Zraster imaging processing is transparently done for you as part of the standard Windows print process.

- For example when printing through a Windows (or Linux) installed ZXP 3 driver. Customer will install the driver and send print jobs through an application. The driver manages USB or Ethernet communications for the user. The method in this example assumes a windows file based application (File > Print)


Host Computer
File>> Print
Driver
ZRaster
Windows Based Application
Image Data Compression Algorithm
Raw Print Data
Print Head Applies Data to card
Card Movement and Sensor information
Printed Card
ZXP 3 Printer

Appendix C

Print Process for ZXP 7 printers.

ZXP 7 Print Process:

The ZXP 7 uses a new modern printer communication protocol architecture called ZMotif Printer control suite which enables the transfer of data between a host computer and a Zebra ZXP 7 printer. All ZXP S7 printer interfaces such as our ZMotif SDK and Windows Drivers, etc…, use and are built on the ZMotif protocol.

The ZMotif protocol is a job oriented command suite and contains Print job control commands as well as commands to transfer data, configure the printer and return information from the printer.

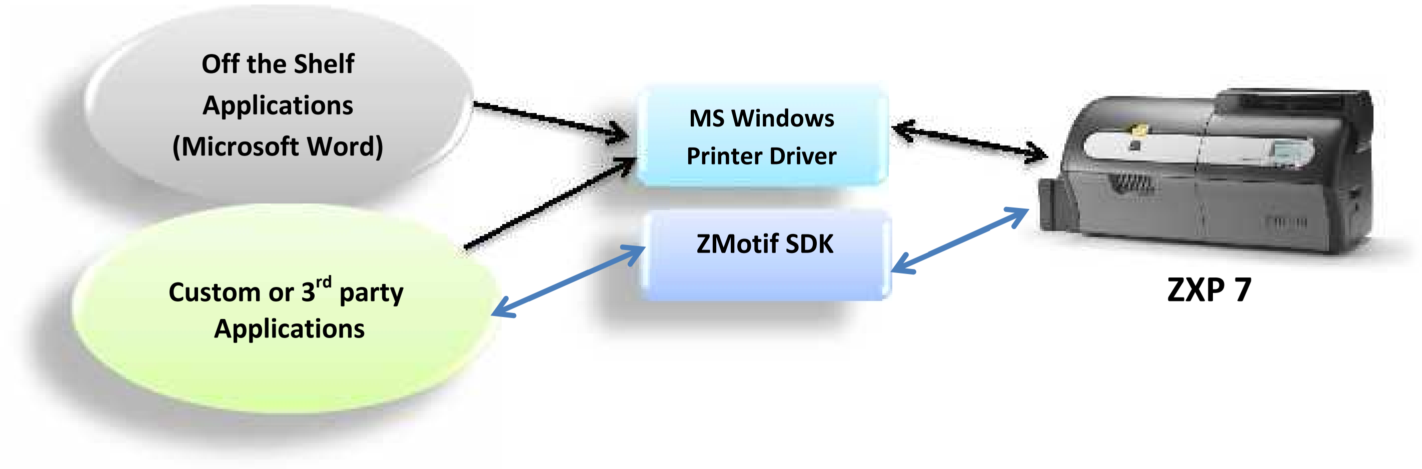
ZMotif control suite jobs are written in XML and transferred to the ZXP 7 printer.

ZMotif comprises two distinct, independent layers:

1. ZMC, ZMotif real-time Communications protocol that handles commands to, and Responses from, the target printer.
2. ZMJ, a ZMotif print Job description protocol that sits above the ZMC communications layer.

The ZMotif is a hardware device dependent low level protocol that requires considerable knowledge of how print jobs and graphics sent to the printer must be formatted and sequenced. To facilitate easy third party application integration of ZXP S7 card printing the ZMotif SDK was developed to create a high level Print Job oriented interface for developers to interface print operations into their applications.

ZMotif Printers (ZXP 7 or 8) can be used in a variety of solutions and applications. They support both USB and Ethernet via a Printer Driver or ZMotif Software Development Kit (SDK). End users can print to ZXP 7 or 8 via off the shelf products, such as Microsoft Word, through the Printer Driver or custom applications using the ZMotif SDK. Additionally, developers can utilize a combination of ZMotif SDK and printer driver together. The ZMotif SDK supports most Windows based development platforms such as .Net and C#. In addition there is a JAVA based SDK version that supports Windows and Linux JAVA based development environments.


Off the Shelf Applications (Microsoft Word)
MS Windows Printer Driver
ZMotif SDK
Custom or 3rd party Applications
ZXP 7