ENTRUST

Sigma Series Card Printers

OpenCard Data Format Guide

March 2021

528222-001, Rev. A

Notice

Please do not attempt to operate or repair this equipment without adequate training. Any use, operation or repair you perform that is not in accordance with the information contained in this documentation is at your own risk.

Trademark Acknowledgments

Entrust, Sigma and the hexagon design are trademarks, registered trademarks and/or service marks of the Entrust Corporation in the United States and other countries.

Datacard is a registered trademark and service mark of Entrust Corporation in the United States and other countries.

MasterCard is a registered trademark of MasterCard International Incorporated.

Visa is a registered trademark of Visa International Service Association.

All other product names are the property of their respective owners.

Proprietary Notice

The design and information contained in these materials are protected by US and international copyright laws.

All drawings and information herein are the property of Entrust Corporation. All unauthorized use and reproduction is prohibited.

Entrust Corporation
1187 Park Place
Minneapolis, MN 55379
Phone: 952-933-1223
Fax: 952-933-7971
www.entrust.com

Revision Log

Revision	Date	Description of Changes
A	March 2021	First release of this document

Table of Contents

Chapter 1: About this Manual

This chapter covers how this guide is organized, who should read this guide, and the conventions this guide uses.

This manual details system requirements, formatting, and printing features of OpenCard Data Format for the Entrust® Sigma Series Card Printers. Only Sigma DS3 Card Printers support printing using OpenCard.

Organization

Information in this manual is organized as follows:

- Chapter 1: "About this Manual"—Contains information about this manual's organization, its intended audience, and the conventions it uses.
- Chapter 2: "Getting Started"—Contains an overview of using OpenCard Data Format with the Sigma Series card printers and the Legacy Option for SP/CP Series printers. It includes requirements for using OpenCard and initial setup procedures.
- Chapter 3: "Creating Card Formats"—Includes the information about how to create card formats for use with Sigma Series card printers.
- Chapter 4: "OpenCard Commands"—Describes the commands that are used in an OpenCard data stream.
- Chapter 5: "Exporting Legacy Card Layouts, Graphics, and Fonts"—Includes information and procedures for using existing card layouts originally created for SP/CP Series printers to print on Sigma Series card printers with OpenCard.
- Chapter 6: "Working with Printer Dashboard"—Explains how to set up the printer to work with OpenCard card formats.
- Chapter 7: "Printing Cards"—Provides information about printing cards on Sigma Series card printers with OpenCard.
- Appendix A: "Sample Card Formats"—Provides working samples of card formats.
- Appendix B: "Setting Up OS/400"—Provides information about how to set up an existing OS/400 system to work with OpenCard.

Intended Audience

This manual is intended primarily for personnel who create and manage files for card formats and data streams, referred to in this guide as card format designers.

Card format designers should have the following knowledge:

- Have a general idea of the appearance and content of the card formats they are defining.
- Understand the type of data to be printed on the cards (text, bar codes, logos).
- Be familiar with printer ribbons and ribbon panels (such as YMCKT ribbon panels).
- Be familiar with the source of the data to be printed on cards.
- Know how to use a command line, browser application, and how to work over a network.

The card format designer's job responsibilities using OpenCard include:

- Work with the Datacard Printer Dashboard interface menus to view and manage posted card format files, images, fonts, and font families.
- Use the Legacy Option to create to card layouts for printing on Datacard SP55, SP60, CP60, and FP65 model printers and export these card layouts using Telnet. These designers use Printer dashboard to import card layouts, fonts, and images to the Sigma Series card printers.
- Use the Sigma Series card printers to create card formats in Scalable Vector Graphics (SVG) markup language and use Printer Dashboard to import card formats, fonts, and images to the Sigma Series card printers.

Conventions this Manual Uses

This manual uses the following conventions:

Convention	Meaning
Sigma Series	Refers to Entrust Sigma DS1, DS2, and DS3 card printers.
Legacy Option	Refers to Datacard SP/CP Series printers, including the SP55, SP60, CP60, and FP65 card printers.
`<1234567890>`	Text displayed in this manual using the style at left is data stream content. Braces display at the start (<) and end (>), and text (1234567890) indicates data stream text.
Card layout	Refers to the card layout files originally created for use with Legacy SP55, SP60, CP60, or FP65 (SP/CP Series) card printers. Used to describe options and procedures for the OpenCard Legacy Option. For more about the original card layouts for printers, refer to the *Data Formatting Guide*.
Card format	Document that provides the instructions for card appearance and content. Uses Scalable Vector Graphics (SVG) standard markup language.

Related Publications

For details about the data stream, Telnet, or the original card layout file created for SP/CP Series card printers, refer to the *OpenCard Data Formatting Guide* (part number 539397-001 Rev D, dated November 2007). That manual supports the SP55, SP60, CP60, or FP65 card printers.

Chapter 2: Getting Started

2

The chapter provides an overview of using the Sigma Series printers with OpenCard Data Format, as well as an overview of the Legacy Option for converting existing card layouts from SP/CP Series printer card layouts. It also explains the requirements for using OpenCard and initial start up procedures.

The Sigma Series card printers with OpenCard Data Format provide the following options:

- **Sigma Series OpenCard Formatting** —Provides enhanced text and graphics capabilities. Create a card format file in Scalable Vector Graphics (SVG) markup language, and import it for use with Sigma Series card printers.
- **Legacy Option for Using SP/CP Series Card Layouts**—Provides compatibility with card layouts originally created for the Datacard SP55, SP60, CP60, or FP65 card printers with OpenCard. Export existing card layouts originally created for a legacy SP/CP Series card printer and import them to the Sigma Series card printer for use with OpenCard.
- **Printer Dashboard**—Provides a browser-based set of menus for managing print operations for either option listed above. This guide provides an introduction to Printer Dashboard features relevant to OpenCard.

Sigma-Series OpenCard Formatting

For Sigma Series card printers, create card formats saved as Scalable Vector Graphics (SVG) documents with any text editor or SVG-specific tool, such as SVG-edit. Using these card formats plus the data stream, OpenCard extracts text and magnetic stripe data from the data stream and places this data on cards according to the appearance and location specifications you define in the card format. Rotate text, graphics, and bar codes individually, or rotate the entire design.

Card formats designed with the Sigma Series formatting are not supported on SP/CP Series printers.

The Sigma Series card printers option includes:

- Card formats written in SVG markup language
- Support ASCII data streams. Any host computer that can generate an ASCII data stream and send it to a direct socket connection can send card data using OpenCard. (The printer does not support the Extended Binary Coded Decimal Interchange [EBCDIC] character set.)
- Support tactile impression elements using the Tactile Impression Module with Sigma Series card printers.

- Support for any operating system, including Windows, Linux, UNIX.
- Front and back monochrome or color printing, and front and back topcoat.
- ISO magnetic stripe tracks 1–3.
- Support for manual card insertion at prompting.
- Support for TrueType fonts and bar code fonts stored on the printer.

  Use text font typefaces that include, but are not limited to, typefaces within font families for Courier, Sans, and Serif, which come bundled with the printer. Define any size for fonts, within the practical limits of the card size and the font families installed on the printer.

  Bar code font typefaces are bundled with the printer. They include but are not limited to:

  - Code39
  - Code128
  - UPCA
  - EAN8
  - EAN13

- Support for a wide variety of image formats stored on the printer, including:

  - JPEG
  - SVG
  - MNG
  - TGA
  - PNG
  - TIFF
  - PBM
  - XBM
  - PGM
  - XPM
  - PPM

Legacy Option for Card Layouts from SP/CP Series Printers

If your organization currently sends data tagged with OpenCard commands to an SP/CP Series card printer or embosser, you can use the same data stream to produce cards using the Legacy Option features shown in this guide. With the Legacy Option, you export the card layouts originally created for a SP/CP Series card printer, and then import those card layouts and their graphics and fonts to the Sigma Series card printer.

The Legacy Option supports OpenCard production data formatting containing text and magnetic stripe data items, as well as card layout selection.

Using the Legacy Option with OpenCard, legacy card layouts maintain their original capabilities:

- Front-side monochrome black printing of text, bar codes, and graphics on a single-sided, landscape-oriented card.
- Import up to four black-and-white image files to the printer. The printer does not provide the ability to download a different logo or graphic for each card.

A note about printing: The Legacy Option supports monochrome printing on one side of the card. On color printers, if you use a full-color ribbon in the printer when you send OpenCard data, the printer uses only the K panel in the ribbon. The color panels are not used.

- ISO magnetic stripe tracks 1, 2 and 3
- Printer-resident fonts, font-families and bar codes

  To use the Legacy Option to produce cards on the Sigma Seriescard printers, you must use text sizes and font typefaces that are available on the Legacy Series printers. These typefaces are within font families for Courier, Sans, and Serif, and are installed on Sigma Series card printers.

  Bar code font typefaces are also bundled with the Legacy Series printers. They include the following:

  - Code39
  - Code128
  - UPCA
  - EAN8
  - EAN13

Printer Dashboard Tools

Printer Dashboard is a web interface hosted on the printer used for configuring and managing print jobs at the printer. Use Printer Dashboard to:

- Import Scalable Vector Graphics (SVG) card formats or SP/CP Series printers' legacy OpenCard card layouts to the printer
- Import images used in the card layout to the printer
- Manage card formats, fonts, and card stocks

OpenCard Requirements

The following components must be in place to use the Sigma Series card printers with OpenCard Data Format.

Printer Requirements

Printing with OpenCard requires Sigma DS3 Card Printers with the OpenCard Data Format option enabled.

Network Requirements

Access or use these printers on networks set up with:

- Computers that are using Transmission Control Protocol/Internet Protocol (TCP/IP) communications. OpenCard supports IPv4 and IPv6 network addresses.
- Web browsers installed on networked computers.

Other Requirements

- If your organization uses the Legacy Option, you need a Telnet interface that provides card layout menus for exporting existing SP/CP Series card layouts.
- For additional information on system set up requirements, except as noted in this guide, refer to your printer's *Installation and Administrator's Guide*.

- Some printer, network, and system setup requirements for Sigma Seriescard printers are overridden by the specifications in this guide.
- Sigma Series card printers that have OpenCard enabled are not recognized or supported by the XPS Card Printer Driver.

Enabling OpenCard

The OpenCard option is active on the printer upon shipment if the printer was ordered with that option. If the OpenCard option is not active, activate it using the OpenCard Upgrade Kit. Perform these three steps to enable OpenCard on the printer:

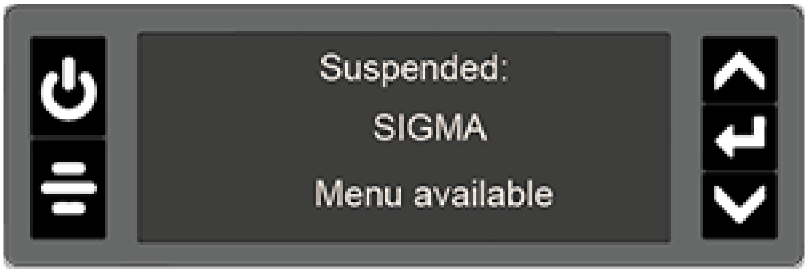
1. Activate OpenCard on page 9
2. Enable OpenCard using the Printer's LCD Menu on page 9
3. Enable OpenCard with Printer Dashboard on page 10

Activate OpenCard

If you ordered the printer with OpenCard, it is already activated. If OpenCard is not activated on the printer, order the OpenCard Upgrade Kit. Then, follow the instructions provided with the kit to activate OpenCard.

Enable OpenCard using the Printer's LCD Menu

Use the following procedure to enable OpenCard for Sigma Series printers using the front panel LCD menu.



Suspended:
SIGMA
Menu available

1. Press the Menu button ≡. Then press the Enter button ↵. The menu displays.
2. Press the down arrow until **Configuration** displays. Then press the Enter button ↵. The Configuration menu displays.
3. Press the down arrow button ⌄ until **OpenCard** displays. Then press the Enter button ↵.
4. Press the down arrow button ⌄ until **enable** displays. Then press the Enter button ↵. The printer enables OpenCard.

Enable OpenCard with Printer Dashboard

To enable OpenCard with Printer Dashboard, you must have a Printer Dashboard account with administrator-level access. Refer to your printer’s *Installation and Administrator’s Guide* for more information about assigning Printer Dashboard access levels. Before you start, make sure that your printer meets all requirements. Refer to “OpenCard Requirements” on page 8.

1. Open Printer Dashboard.
    1. Use the printer’s LCD menu to get the printer IP address. Refer to the printer’s User’s Guide for information.
    2. Open a web browser on the computer then enter the following address:

        https://*[printer IP address]*/

        The browser displays a certificate warning because it is a secure connection.
    3. Click **Continue to this website (not recommended)**. The Log In page displays.
    4. Enter a **User ID** and **Password** then click **Login**. The Printer Dashboard opens.
2. In Printer Dashboard, select Main Menu ☰ **> Configuration > Settings**. The Settings page opens.
3. From the drop-down list below Change Settings, select **Behavior**.
4. Set the value for the **Plugin** setting to **Enabled**.
5. Click **Save**.
6. Restart the printer.
    1. Select Main Menu ☰ **> Troubleshooting > Restart Printer**. The Restart Printer page opens.
    2. Click **Restart**. The printer restarts.

Enable OpenCard Legacy Mode with Printer Dashboard

OpenCard Legacy Mode enables the printer to process card formats originally created for SP/CP Series card printers. Before you start, make sure that your printer meets all requirements. Refer to "OpenCard Requirements" on page 8.

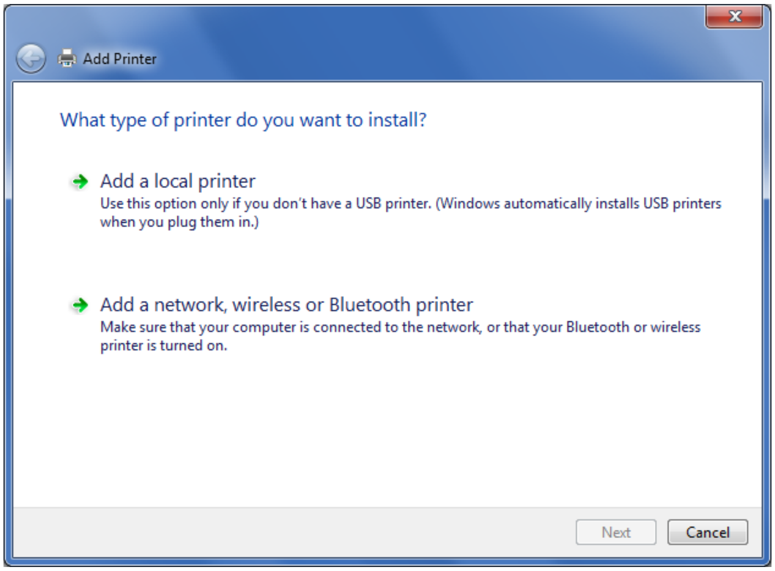
1. In Printer Dashboard, select Main Menu ☰ **> Configuration > Settings**. The Settings page opens.
2. From the drop-down list below Change Settings, select **Behavior**.
3. Change the setting for **OpenCardLegacyMode** to **Enabled**.
4. Click **Save**.
5. Restart the printer.
    a. Select Main Menu ☰ **> Troubleshooting > Restart Printer**. The Restart Printer page opens.
    b. Click **Restart**. The printer restarts.

Setting Up a Generic/Text Driver with OpenCard Printers

Use this procedure for Windows applications that print using OpenCard format.

Generic/Text Driver for Networks

1. Open **Devices and Printers** from the Control Panel.
2. Click **Add a printer**. The Add Printer wizard opens.
3. Select **Add a local printer**.



What type of printer do you want to install?

Add a local printer

Use this option only if you don't have a USB printer. (Windows automatically installs USB printers when you plug them in.)

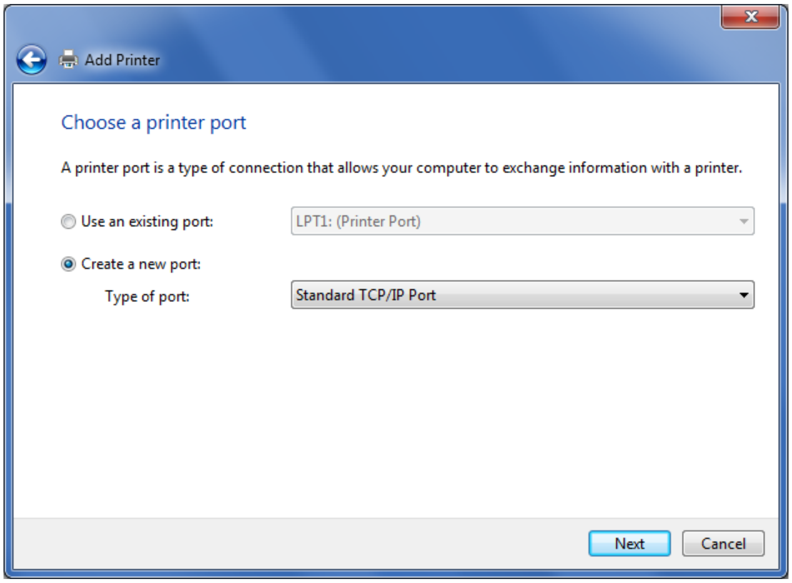
Add a network, wireless or Bluetooth printer

Make sure that your computer is connected to the network, or that your Bluetooth or wireless printer is turned on.

4. Set up a new TCP/IP port.

   a. Select **Create a new port** on the Choose a printer port window.

   b. Select **Standard TCP/IP Port** from the drop down menu.



Add Printer

Choose a printer port

A printer port is a type of connection that allows your computer to exchange information with a printer.

○ Use an existing port: LPT1: (Printer Port)

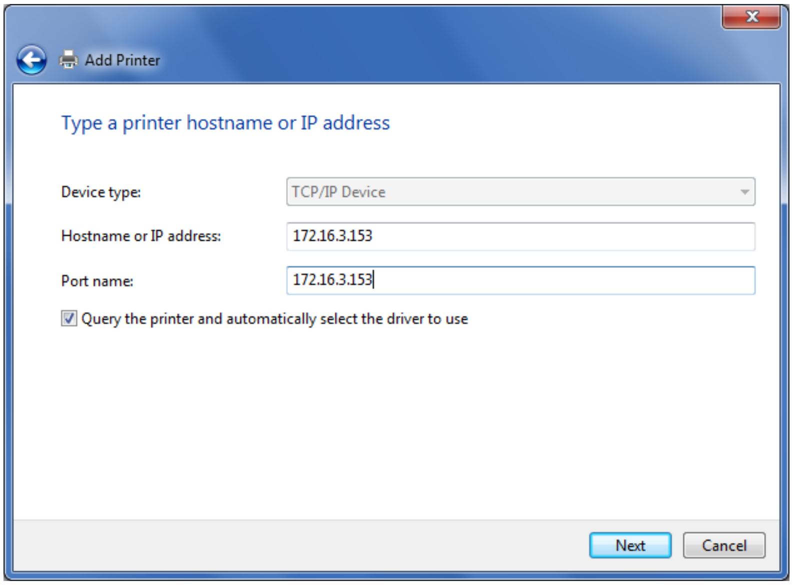
● Create a new port:

Type of port: Standard TCP/IP Port

   c. Click **Next**.

5. In the Hostname or IP address text box, enter the IP address of the printer. Sigma Series card printers support both IPv4 and IPv6 addresses.

6. Click **Next**.



Add Printer

Type a printer hostname or IP address

Device type: TCP/IP Device

Hostname or IP address: 172.16.3.153

Port name: 172.16.3.153

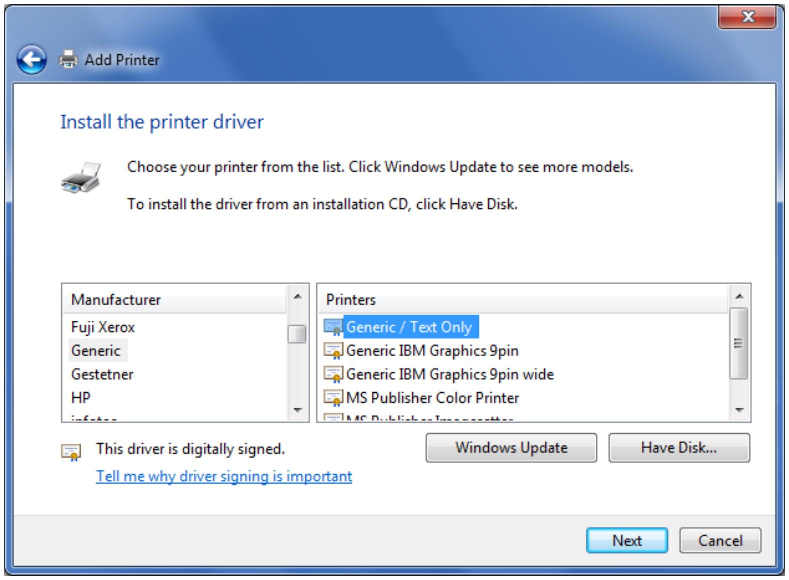
☑ Query the printer and automatically select the driver to use

7. Configure the generic driver.

   a. In the Manufacturer list, click **Generic**.

   b. In the Printers list, click **Generic/Text Only** .

   c. Click **Next**.



Add Printer

Install the printer driver

Choose your printer from the list. Click Windows Update to see more models.

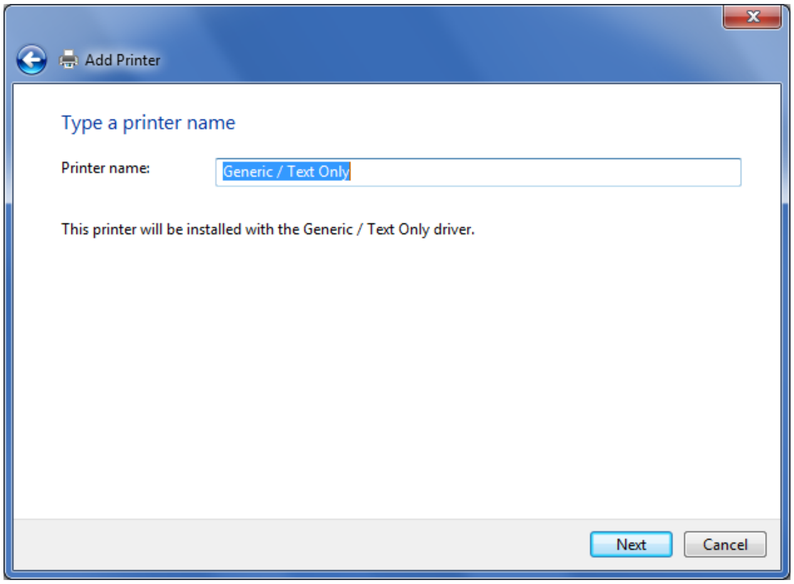
To install the driver from an installation CD, click Have Disk.

Manufacturer	Printers
Fuji Xerox	Generic / Text Only
Generic	Generic IBM Graphics 9pin
Gestetner	Generic IBM Graphics 9pin wide
HP	MS Publisher Color Printer

This driver is digitally signed.

Tell me why driver signing is important

8. In the Printer Name field, enter **Generic / Text Only**. Then, click **Next**.



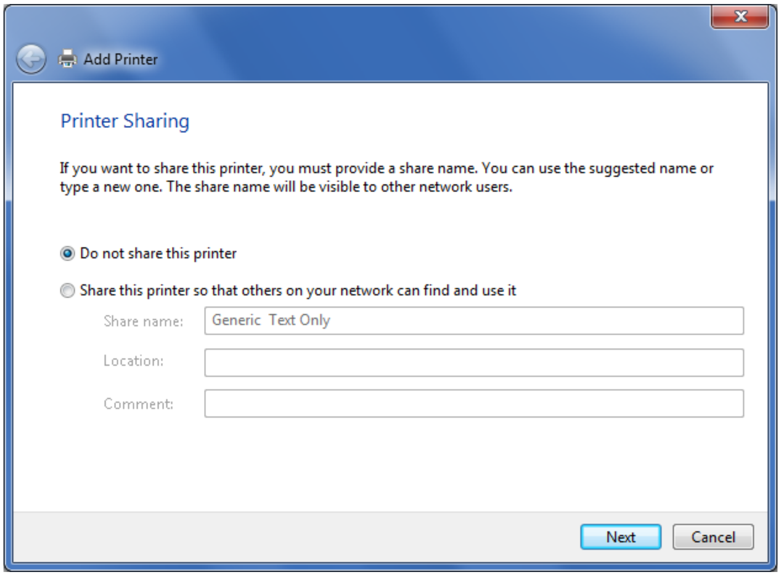
Add Printer

Type a printer name

Printer name: Generic / Text Only

This printer will be installed with the Generic / Text Only driver.

9. Select **Do not share this printer**. Then, click **Next**.



Printer Sharing

If you want to share this printer, you must provide a share name. You can use the suggested name or type a new one. The share name will be visible to other network users.

● Do not share this printer

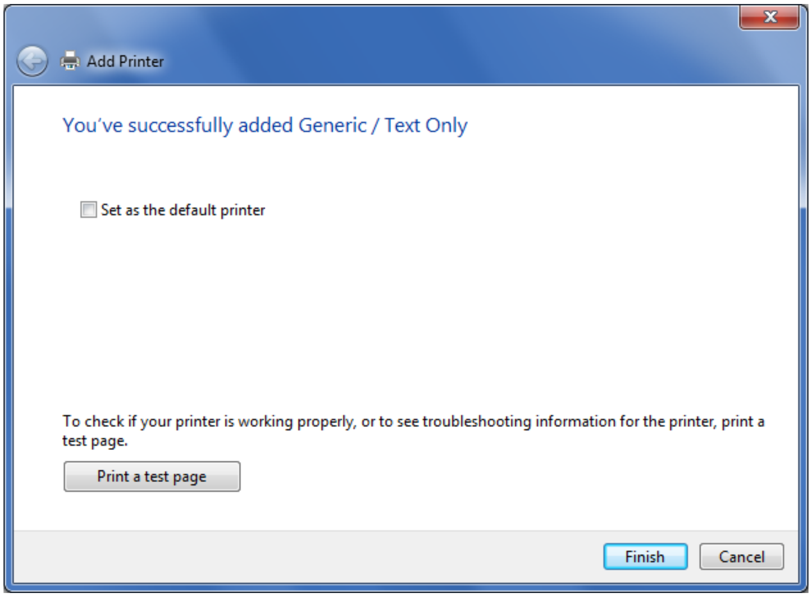
○ Share this printer so that others on your network can find and use it

Share name: Generic Text Only

Location:

Comment:

10. Click **Finish**.



You've successfully added Generic / Text Only

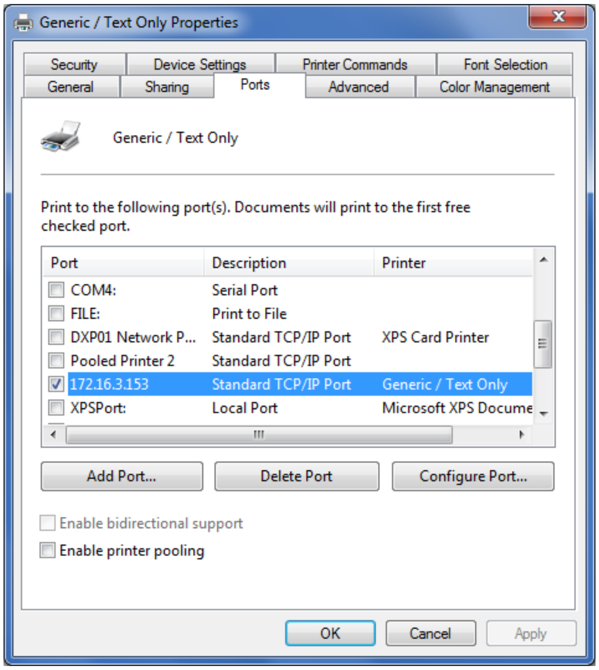
□ Set as the default printer

To check if your printer is working properly, or to see troubleshooting information for the printer, print a test page.

11. Verify Generic / Text Only Properties

    a. Right-click the printer name on the Devices and Printers window and select **Printer properties**.

    b. Click the Ports tab and verify that the IP address is correct.



Generic / Text Only

Print to the following port(s). Documents will print to the first free checked port.

Port	Description	Printer
☐ COM4:	Serial Port	
☐ FILE:	Print to File	
☐ DXP01 Network P...	Standard TCP/IP Port	XPS Card Printer
☐ Pooled Printer 2	Standard TCP/IP Port	
☑ 172.16.3.153	Standard TCP/IP Port	Generic / Text Only
☐ XPSPort:	Local Port	Microsoft XPS Docume

☐ Enable bidirectional support

☐ Enable printer pooling

Install Generic/Text Driver for USB Connection

1. Install the XPS Printer Driver on the USB port.
2. Retrieve the USB IP address from the printer.
3. Install the Windows Generic Driver and mount the IP address to the USB port.
4. Make sure that the Windows Generic Driver is set to use RAW mode.

Chapter 3: Creating Card Formats

3

This chapter presents an OpenCard process overview, and describes how to create an OpenCard card format to use with the Sigma Series card printers.

Card Formats

The card format defines personalization operations on both the front and back side of a card. OpenCard supports monochrome and color graphics, topcoat and printing on both sides of the card, as well as magnetic stripe encoding and tactile impression elements. Enhanced capabilities include support for a full range of industry-standard graphics image formats and any TrueType font for text. In addition, you can define a wide range of image and text transformations.

Card Personalization Process Overview

Card formats merge with the OpenCard data stream to create a personalized card using the following process:

1. **Setup**

   The card format designer creates a card format as an SVG document, creates or acquires the images and fonts called out in the card format, and prepares the data stream. The data stream is a text stream that can contain text lines, an optional magnetic stripe data line, and command lines. Preparing the data stream also includes optionally naming a specific card format with the @G command, which indicates the card format to use, and naming the card stock with the @C command.

2. **Import**

   The card format designer uses Printer Dashboard to import the card format, images, and fonts to any Sigma Series printer that is to use the card format.

3. **Production**

   - The printer receives an OpenCard data stream.
   - OpenCard locates the correct card format on the printer.

- OpenCard uses the instructions in the card format to merge the data stream text, images, and magnetic stripe data into a card to be printed, similar to a mail merge. As in the mail merge process, the card format contains text and image named variables that OpenCard uses to map to the corresponding items in the data stream.

  When a printer receives OpenCard commands with no active card layout on the printer, it prints using the Default card layout file. If there is no active card layout and the printer cannot find the Default card layout file, printing fails and the printer shows an error. To prevent the error, create a card layout file with the filename Default (no file extension).

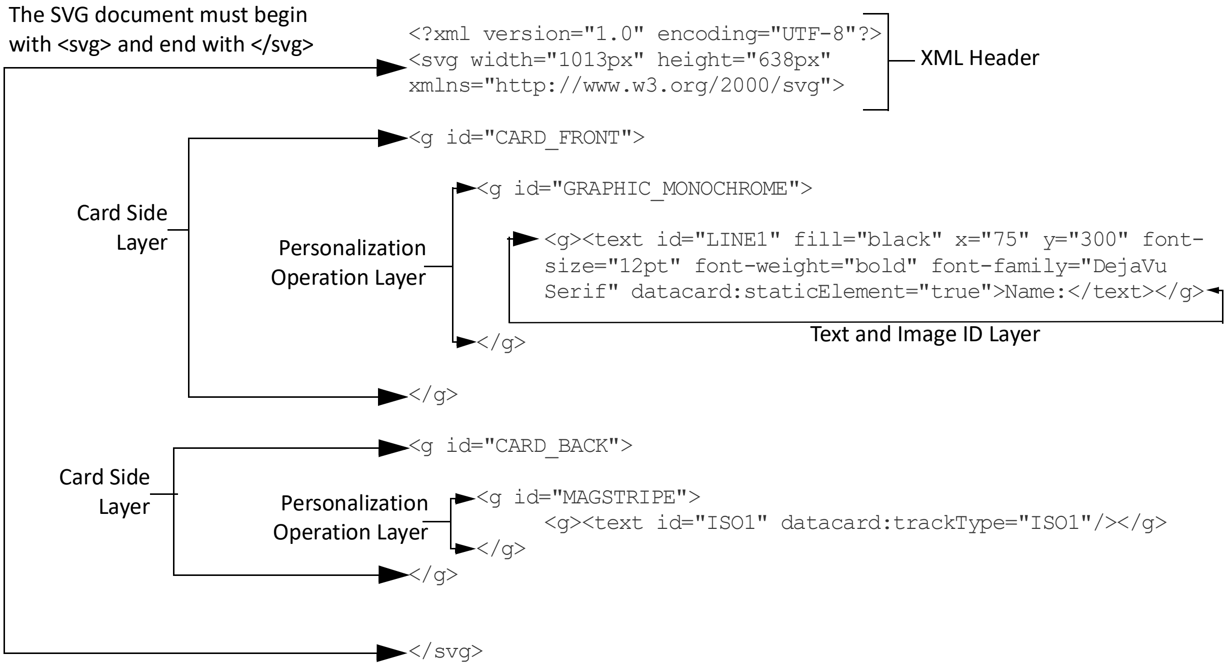
The Card Format as an SVG Document

The card format is an XML-based document, Scalable Vector Graphics, or SVG. To create an OpenCard card format, the SVG standard defines the basic structure of the card format, and the majority of the attributes that define image and text items.

The SVG card format also uses Entrust-specific conventions and extensions compatible with the OpenCard data stream standard, affording access to the full range of capabilities available in Sigma Series printers. For information about the OpenCard data stream standard, refer to the Data Formatting Guide.

Card Format Structure and Conventions

The SVG markup you use to create an OpenCard card format describes the relationships between the elements that make up the card format as layers. Each layer is demarcated with a <g> at the beginning and a </g> at the end, as shown in the following example:


The SVG document must begin with <svg> and end with </svg>
XML Header
Card Side Layer
Personalization Operation Layer
Text and Image ID Layer
Card Side Layer
Personalization Operation Layer

```
<?xml version="1.0" encoding="UTF-8"?>
<svg width="1013px" height="638px"
xmlns="http://www.w3.org/2000/svg">
<g id="CARD_FRONT">
   <g id="GRAPHIC_MONOCHROME">
      <g><text id="LINE1" fill="black" x="75" y="300" font-
      size="12pt" font-weight="bold" font-family="DejaVu
      Serif" datacard:staticElement="true">Name:</text></g>
   </g>
</g>
<g id="CARD_BACK">
   <g id="MAGSTRIPE">
      <g><text id="ISO1" datacard:trackType="ISO1"/></g>
   </g>
</g>
</svg>
```

XML Header—The first line of an OpenCard card format must be a standard XML declaration. The header must also include the SVG declaration shown, with the beginning <svg> and its corresponding </svg> at the end of the card format. An XML header also includes the card dimensions, which correspond to the number of pixels (px) when printing at 300 dpi (width = 3.375” x 300 = 1013 px; height = 2.125” x 300 = 638 px). This is the only resolution supported.

Card Side Layer—Identifies on which side of the card the personalization occurs.

datacard:translations (not shown)—Applied at the same level as a card side layer, datacard:translations are instructions to apply translations to the data stream.

Personalization Operation—Defines monochrome or color graphics, and instructions for magnetic stripe and topcoat.

datacard:flip (not shown)—Applied at the same level as the personalization operation, datacard:flip flips the card to prepare it for personalization.

Text and Image ID Layers—Provide the appearance and placement details for text and images.

Naming a Card Format in the Data Stream

The @G command in the OpenCard data stream defines which card format (or Legacy SP/CP Series card layout) to use to print the card.

- If an @G command is in the data stream, then the printer uses the card format. If the named card format is not loaded on the printer, then the card is rejected and does not print.
- If no @G command exists in the data stream, then the printer software uses the card format named Default. If Default doesn't exist, the card is rejected and the request does not print.

Naming Card Stock in the Data Stream

The @C command in the OpenCard data stream defines which card stock to use. A card stock definition is required when setting up the LCD panel to prompt the user to insert a card in the exception slot of the input hopper. Prompts from the LCD panel are shown only with manual card insertion.

- If an @C command is in the data stream, then the printer uses the card stock requested.
- If the card stock is not defined on the printer, then the printer uses the card stock defined in Printer Dashboard as Default. If Default doesn't exist, the card is rejected and the request does not print.

ID Naming Conventions

Data	ID Naming Conventions
Dynamic Text Data	To personalize a text line of data, the ID convention is LINE*n*, where *n* can be from 1 to 15. An ID of LINE1 then uses data supplied in the first personalization data line of an OpenCard data stream.
Magnetic Stripe Data	To place magnetic stripe data, the magnetic stripe data line must use an ID that conforms to the convention ISO*n*, where *n* can be 1, 2, or 3—corresponding to ISO tracks 1, 2, and 3. (Alternatively, when there are no pre-defined magnetic stripe commands in the data stream, you can use a dynamic text ID of LINE*n* to identify which line text data to place on a specific track.)
Tactile Impression Data	To add a tactile impression element to the card, identify tactile impression data using an ID that follows this format: IMP*n* where *n* represents the number of the tactile impression element. Currently, Sigma Series printers support only one tactile impression element per card.
Static Data	Static data is data that does not derive its images or text from the OpenCard data stream. Since the data is the same every time, it can be defined in the card format itself. Static data does not have to have an associated ID. If it does have an ID, you can use any name other than the IDs "LINE*n*", "ISO*n*", or "IMP*n*" reserved for dynamic date, magnetic stripe data, and tactile impression data respectively as described above.

XML Escape Characters

Escape characters are characters in XML that identify XML code. For example, < identifies the start of a new tag in XML. Occasionally, escape characters must be included in the data and not be interpreted as XML code. To use escape characters in the card format data, replace the escape characters with the characters listed in the table below.

Character Name	Escape Character	Replacement Character
Quotation Mark	“	&quot;
Apostrophe	‘	&apos;
Less Than	<	&lt;
Greater Than	>	&gt;
Ampersand	&	&amp;

Adding Elements to the Card Format

This section provides information about how to apply the elements available to create an OpenCard card format. Elements are listed in the order they are applied in the SVG card format document.

Card Side Layers

Within the SVG document, a card format must define a front side layer and/or a back side layer.

- Use an ID of “CARD_FRONT” or “CARD_BACK”.
- Add personalization layers as children of each of the front and back side layers as needed.
- “CARD_FRONT” and “CARD_BACK” are the only layer IDs OpenCard recognizes at this level.

Card Side Layers Example

```
<?xml version="1.0" encoding="UTF-8"?>
<svg width="1013px" height="638px" xmlns="http://www.w3.org/2000/svg">

       <g id="CARD_FRONT">
       ... front side card personalization operations go here ...
       </g>

       <g id="CARD_BACK">
       ... back side card personalization operations go here ...
       </g>
```

```
</svg>
```

Re-Mapping Data Characters with datacard:translations

Translations convert data streams. Standard character translations convert one character to another. Advanced translations can convert characters or strings of characters within the data or over the entire data stream. Using this element, the card format designer can define a list of characters to translate.

The datacard:translations element is placed at the same level as the card side layers because translations are applied to the data stream before it is merged into the card format. Translations affect the value of the data merged into any operation defined in the card format.

Hexadecimal Notation for Translations

For datacard:translations, control characters with values of less than 0x20 (ASCII SPACE character) must be encoded in hexadecimal notation using a leading 0x. For example, represent the null character as 0x00. If a translation maps any data stream character to a null (0x00), then the presence of the null serves to end the data line at that point. Any data following the character that translates to null is not merged into the card format.

Standard Translation

Standard character translations replace one character with another character throughout the data stream.

The following is an example that signals the printer software to translate:

- Any uppercase “A” character to a lowercase “a”
- Any lowercase “a” to an uppercase “A”
- Any uppercase “B” character to a null (0x00)

OpenCard data stream	`<123AaB9876>`

Card format	<?xml version=”1.0” encoding=”UTF-8”?> <svg width=”1013px” height=”638px” xmlns=”http://www.w3.org/2000/svg”> <datacard:translations> <datacard:translate from=”A” to=”a”/> <datacard:translate from=”a” to=”A”/> <datacard:translate from=”B” to=”0x00”/> </datacard:translations> <g id=”CARD_FRONT”> ... front side card personalization operations go here ... </g> <g id=”CARD_BACK”> ... back side card personalization operations go here ... </g> </svg>

Data merged into card format	`LINE1=123aA`

Advanced Translations

Advanced translations provide more complex and powerful options for converting data streams. To enable advanced translations, add the `type` attribute to the `datacard:translate` command.

Advanced translations cannot be used in conjunction with standard translations. If the card format contains both standard translations and advanced translations, the standard translations will operate as advanced character translations. Full stream translations are performed before all other translations regardless of their placement in the card format. The remaining advanced translations are performed in the order they appear in the card format file.

Refer to the examples in this section to enable advanced translations.

Character Translation

Character translation is similar to standard translation but it also translates commands. Standard translations do not translate characters following and including an @ symbol. Also, OpenCard processes character translations before the standard translations. To enable character translation, add the following to the `datacard:translate` command:

```
type=”char”
```

Character Translation Example

In this example, the translation replaces “Z” with “z”, “z” with “Z”, then “b” with “C”.

OpenCard data stream

```
<z34
Z$bg
@GtranslateAdv1.svg>
```

Card format

```
<?xml version="1.0" encoding="UTF-8"?>
<svg width="1013px" height="638px" xmlns="http://www.w3.org/2000/svg">
<datacard:translations>
<datacard:translate from="Z" to="z" type="char"/>
<datacard:translate from="z" to="Z"/>
<datacard:translate from="b" to="C" type="char"/>
</datacard:translations>
<g id="CARD_FRONT">
... front side card personalization operations go here ...
</g>
<g id="CARD_BACK">
... back side card personalization operations go here ...
</g>
</svg>
```

Data merged into card format	`LINE1=Z34`
	`LINE2=Z$Cg`

String Translation

String Translations allow you to replace entire strings of data in the data stream. To enable string translation, add the following to the `datacard:translate` tag:

```
type=”string”
```

To include special characters or formatting in the replaced data, use the following escape sequences:

Escape Sequence	Definition
\\	Backslash
\’	Apostrophe
\”	Quotation Mark
\n	New Line
\r	Carriage Return
\t	Tab indent
\xhh	ASCII Character with hex value

String Translation Example

In this example, character translation replaces % with $, string translation modifies two lines to become one line, and regular expression translation removes data in parenthesis, as well as spaces just prior to the parenthesis if they exist.

OpenCard data stream	`<Johns Widgets[CRLF]`
	`John User (Senior Associate)[CRLF]`
	`%[CRLF]`
	`25[CRLF]`
	`@GtranslateAdv2.svg>`

Card format

```
<?xml version="1.0" encoding="UTF-8"?>
<svg width="1013px" height="638px" xmlns="http://www.w3.org/2000/svg">
<datacard:translations>
<datacard:translate from="%" to="$" type="char"/>
<datacard:translate from="$\r\n" to="$" type="string"/>
<datacard:translate from=" *?\(.*?\)" to="" type="regex"/>
</datacard:translations>
<g id="CARD_FRONT">
... front side card personalization operations go here ...
</g>
<g id="CARD_BACK">
... back side card personalization operations go here ...
</g>
</svg>
```

Data merged into card format

```
LINE1=Johns Widgets
LINE2=John User
LINE3=$25
```

Regular Expression Translation

Regular expressions are sequences of characters that initiate a search and replace within set of data. Regular expression translations use regular expressions to search and replace data in the data stream.

To enable the regular expression translation, add the following to the `datacard:translate` tag:

```
type=”regex”
```

Regular expression translations use a set of special characters to perform actions or represent data within the command. The actions of the special characters are labeled as greedy if they return all results that match the regular expression. The actions are lazy when it returns only minimal results. The default behavior of regular expressions is to be greedy. To specify a lazy search, append ? after the special character.

The following are some of the special characters and their definitions:

Special Character	Definition/Action
.	Represents any character except newline
^	Represents the start of a new line
$	Represents the end of a line
*	Represents 0 or more repetitions of the preceding expression (greedy)

Special Character	Definition/Action
+	Represents 1 or more repetitions of the preceding expression (greedy)
?	Represents 0 or 1 repetitions of the preceding expression (greedy)
*?, +?, ??	Performs the same search as ?, +, and * but it limits it to lazy results.
\	When proceeding a character, it acts as an escape character for that character. For example, to search for *, add * to the regular expression translation.
[]	Indicates a set of characters to search.
|	Logical or for regular expressions.

Regular Expression Translation Example:

In this example, string translation replaces “Account Limit” with “Limit” and regular expression translation adds $ by searching for “Limit:” then appending $ to the searched phrase. In the example, \1 adds the search phrase. So, in this example, “\1” adds “Limit:”.

OpenCard data stream

```
<Johns Widgets
John User
Account Limit: 33.00
@GtranslateAdv3.svg>
```

Card format

```
<?xml version="1.0" encoding="UTF-8"?>
<svg width="1013px" height="638px" xmlns="http://www.w3.org/2000/svg">
<datacard:translations>
<datacard:translate from="Account Limit:" to="Limit:" type="string"/>
<datacard:translate from="(Limit: *)" to="\1$" type="regex"/>
</datacard:translations>
<g id="CARD_FRONT">
... front side card personalization operations go here ...
</g>
<g id="CARD_BACK">
... back side card personalization operations go here ...
</g>
</svg>
```

Data merged into card format

```
LINE1= Johns Widgets
LINE2=John User
LINE3=Limit: $33.00
```

Entire Stream Attribute

The entire stream attribute allows advanced translations to be performed over all data in the data stream. They are not limited to data within data commands.Advanced translations with the entire stream attribute are performed before all other translations. The entire stream attribute must be used in conjunction with an advanced translation.

To add the entire stream attribute, add the following to the `datacard:translate` tag along with one of the advanced translation type attributes:

```
entirestream=”true”
```

Entire Stream Attribute Example

In this example, the original data stream contains an invalid hopper selection. Using the string translation with the entire stream attribute, the translation replaces the invalid hopper selection with a compatible card stock selection.

OpenCard data stream

```
=H3>
<
XYZ
Name: JKL
@Gcardformat.svg>
=H3>
<
XYZ
Name: TUV
@Gcardformat.svg>
```

Card format

```
<?xml version="1.0" encoding="UTF-8"?>
<svg width="1013px" height="638px" xmlns="http://www.w3.org/2000/svg">
<datacard:translations>
<datacard:translate from="=H3&gt;\r\n&lt;" to="&lt;@Ccstock3"
type="string" entireStream="true"/>
<datacard:translate from="(Name: *)" to="" type="regex"/>
</datacard:translations>
<g id="CARD_FRONT">
... front side card personalization operations go here ...
</g>
<g id="CARD_BACK">
... back side card personalization operations go here ...
</g>
</svg>
```

Data merged into card format	`<@Ccstock3`
	`XYZ`
	`JKL`
	`@Gcardformat.svg>`
	`<@Ccstock3`
	`XYZ`
	`TUV`
	`@Gcardformat.svg>`

Personalization Operations

Within each of the card side elements, the card format designer is able to create personalization operations, each as an SVG layer. Datacard-specified personalization operations include:

- Defining Monochrome for Text and Graphics on page 31
- Defining Color for Text and Graphics on page 32
- Defining Topcoat Application on page 33
- Defining Magnetic Stripe Tracks on page 34
- This layer also includes: Flipping the Personalization Operation with the datacard:flip Attribute on page 37

To define one of these personalization operations, create an SVG layer with the <g> element and set the ID attribute to one of the personalization operations. End the definition with a </g>.

Defining Monochrome for Text and Graphics

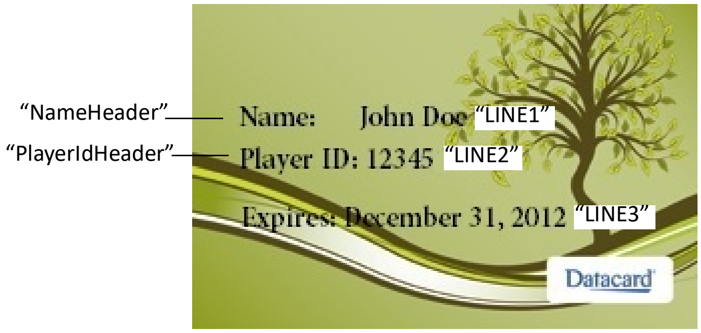
Use the GRAPHIC_MONOCHROME operation to define the text and image elements to print using a monochrome “K” ribbon panel at 300 dpi on either side of the card. All formatting details of text and image elements using monochrome are children of this layer, contained between <g id="GRAPHIC_MONOCHROME"> and its corresponding </g>.

GRAPHIC_MONOCHROME Example

Card format

```
<g id="GRAPHIC_MONOCHROME">

       <g><text id="NameHeader" fill="black" x="75" y="300" font-size="12pt"
                 font-weight="bold" font-family="DejaVu Serif"
                 datacard:staticElement="true">Name:</text>
       </g>


       <g><text id="LINE1" fill="black" x="375" y="300" font-size="12pt"
                 font-family="DejaVu Serif"/>
       </g>


       <g><text id="PlayerIdHeader" fill="black" x="75" y="400"
                 font-size="12pt" font-weight="bold"
                 font-family="DejaVu Serif"
                 datacard:staticElement="true">Player ID:</text>
       </g>


       <g><text id="LINE2" fill="black" x="375" y="400" font-size="12pt"
                 font-family="DejaVu Serif" datacard:format="XXXXX"/>
       </g>


       <g><text id="LINE3" fill="black" x="75" y="525" font-size="11pt"
                 font-family="DejaVu Serif"
                 datacard:appendData="true">Expires </text>
       </g>

</g>
```

Card result



For this example, placing each of these five text IDs within the GRAPHIC_MONOCHROME operation tells the printer to print each of the them with monochrome ribbon.

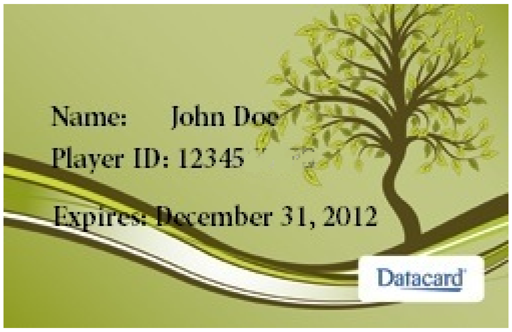
Note that all other details about size, font, placement and data stream source are part of the text ID.

Defining Color for Text and Graphics

The GRAPHIC_COLOR operation defines all of the text and image elements to print using a three-panel combination of yellow, magenta, and cyan (YMC) at 300 dpi on either side of the card. The details of text and image elements are children of this layer, contained between <g id="GRAPHIC_COLOR"> and its corresponding </g>.

GRAPHIC_COLOR Example

Card format	`<g id="GRAPHIC_COLOR">`
	`      <g><image id="Background" datacard:staticElement="true” y="0"`
	`      x="0" height="638px" width="1013px"xlink:href="TreeDebit.jpg" />`
	`      </g>`
	`</g>`

Card result	Any items to be printed in color are defined within the GRAPHIC_COLOR operation. In this example, the background image TreeDebit.jpg requires color processing. Note that the x and y coordinates define the background image placement, and the image dimensions are described in pixels (px), scaled to the full height and width of the card.  Name: John Doe Player ID: 12345 Expires: December 31, 2012 Datacard

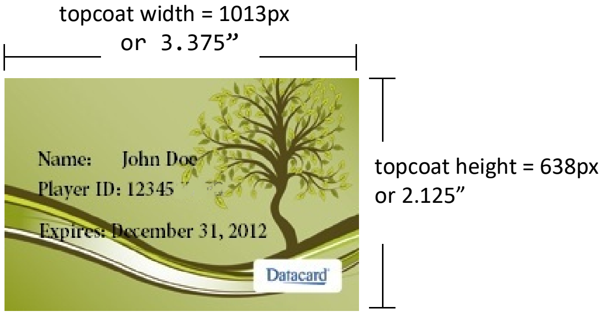
Defining Topcoat Application

The TOPCOAT operation defines the image element to use to apply a topcoat pattern using the T ribbon panel at 300 dpi on either side of the card. The formatting details of image elements are children of this layer, contained between <g id="TOPCOAT"> and its corresponding </g>.

TOPCOAT Example

<table>
<tr>
<td><b>Card format</b></td>
<td><pre>
<g id="TOPCOAT">

       <g><image id="TopcoatImage" datacard:staticElement="true"
                  y="0" x="0"height="638px" width="1013px"
                  xlink:href="Topcoat_Full.png" />
       </g>
</g>
</pre></td>
</tr>
<tr>
<td><b>Card result</b></td>
<td>In this example, the topcoat is applied across the entire width and height of the card 3.375” x 2.125” (or 1013px by 638px)


</td>
</tr>
</table>

Defining Magnetic Stripe Tracks

The MAGSTRIPE operation defines the magnetic stripe tracks to personalize on the front and/or the back side of the card. There are two methods of encoding magnetic stripe tracks:

- Extract text data directly from the data stream and encoding the magnetic stripe using the “LINE*n*” text ID
- Extract magnetic stripe text data already identified with magnetic stripe commands and encoding using the “ISO*n*” text ID

Encoding Text Data to a Magnetic Stripe Track Using the “LINE*n*” Text ID

To populate a magnetic stripe track with text data not originally identified as magnetic stripe data in the data stream, use the “LINE*n*” text ID in conjunction with the datacard:trackType definition, as shown below.

MAGSTRIPE Example 1

```
<g id="MAGSTRIPE">
       g><text id="LINE2" datacard:trackType="ISO1"/></g>
```

Encoding Magnetic Stripe Text on a Magnetic Stripe Track Using the “ISO*n*” Text ID

To populate a magnetic stripe track with prepared magnetic stripe date, OpenCard data stream uses the following magnetic stripe data identifiers:

trackType	Start Sentinel	End Sentinel
ISO1 (IATA)	% (25 hex)	? (3F hex)
ISO2 (ABA)	; (3B hex)	? (3F hex)
ISO3 (TTS)	_ (5F hex) OR _; (5F3B hex)	? (3F hex)

When the OpenCard data stream has text data prepared for magnetic stripe in this way, identify which information encodes to each track using the “ISO*n*” text ID using the following rules:

- The text ID must be “ISO1” for track 1, “ISO2” for track 2, or “ISO3” track 3.
- The datacard:trackType value must match the ID name.

The following example shows how text defined for magnetic stripe in the data stream merges with the card format using these mappings. This merged card format becomes the final instructions for sending the data to the correct track on the magnetic stripe.

MAGSTRIPE Example 2

OpenCard data stream

```
<”%TESTING321?;=1234567890?_;=0987654321?>
```

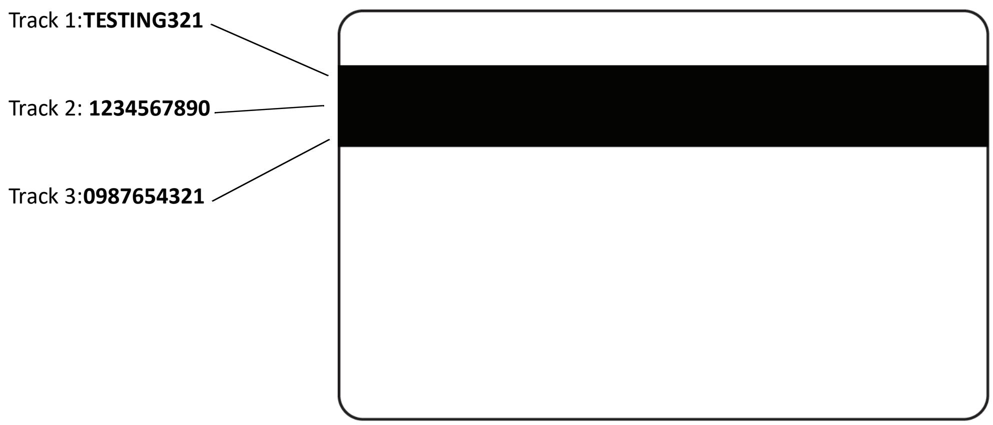
Card format

```
<g id=”MAGSTRIPE”>
       <g><text id=”ISO1” datacard:trackType=”ISO1”></text></g>
       <g><text id=”ISO2” datacard:trackType=”ISO2”></text></g>
       <g><text id=”ISO3” datacard:trackType=”ISO3”></text></g>
</g>
```

Data merged into card format

```
<g id="MAGSTRIPE">
       <g><text id="ISO1"datacard:trackType="ISO1">TESTING321</text></g>
       <g><text id="ISO2"datacard:trackType="ISO2">=1234567890</text></g>
       <g><text id="ISO3"datacard:trackType="ISO3">=0987654321</text></g>

</g>
```

Card result

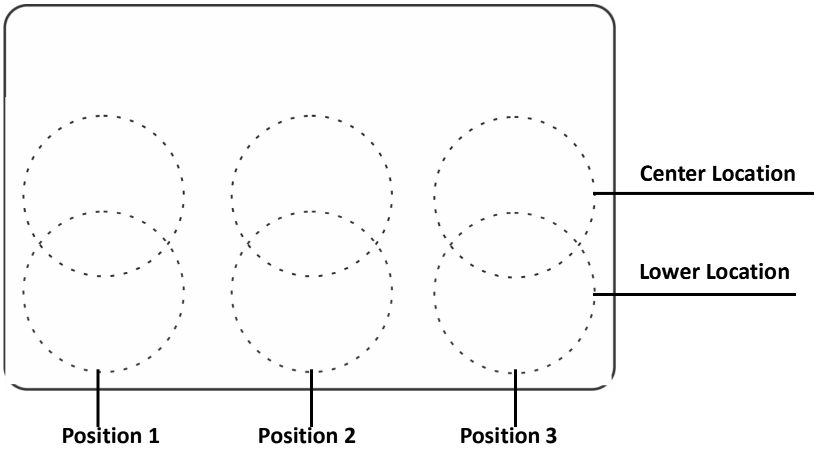


Defining Tactile Impression Elements

The IMPRESS operation enables the Tactile Impression Module (TIM) to imprint a design on the card. The printer can imprint only once per card.

The TIM imprints to one of three horizontal positions on the front or back of the card. Position 1 is closest to the leading edge of the card, position 2 is in the middle, and position 3 is near the trailing edge of the card. The horizontal position is set in the IMPRESS operation. The TIM also imprints to one of two vertical locations. The vertical locations are mechanically set in the TIM.

In this example, the left side of the card is the leading edge of the card.



Follow these guidelines to configure an element for TIM printing.

- The operation layer must be within a card side layer to identify the side on which the element will impress.
- The ID of the element must be `IMP1` to identify that element to be impressed using the TIM.
- The element must be defined with `datacard:staticElement="true"` because the impressed data must be static and cannot be no data stream data will modify the data ti be impressed.
- Optionally, add `datacard:impressPosition` to the element to selected one of three horizontal positions. If no position is selected, the printer uses position one by default.

IMPRESS Example

```
<g style="display:inline" id="CARD_FRONT">
  …
  <g id="IMPRESS">
   <g>
      <text id="IMP1" datacard:staticElement="true" datacard:impressPosition="1"/>
   </g>
  </g>
```

```
</g>
```

Flipping the Personalization Operation with the datacard:flip Attribute

If "true", this attribute causes the entire personalization operation design to be flipped 180 degrees. Valid values are "true" and "false", with the default equal to "false" if the attribute is not present. This attribute does not have any effect on a MAGSTRIPE personalization operation.

datacard:flip Example

```
<g id="GRAPHIC_MONOCHROME" datacard:flip=”true”>
. . . text and image elements go here . . .
</g>
```

Adding Text and Image Layers

Within each personalization operation, you create text and image layers that describe all aspects of the actual text and image, as described in the following:

- Defining Text ID Layers on page 37
- Using Bar Code-Specific Attributes on page 44
- Defining Image Element Layers on page 47

Defining Text ID Layers

Attributes for height, width, font-family, transformations, and other location and appearance details of text data to be personalized to a card are set here. In addition, bar code fonts and other bar code attributes are defined within a text ID layer. Begin each text or image layer with <g>, add the text or graphic elements and their attributes, and then end the definition with </g>.

When used in a GRAPHIC_MONOCHROME or GRAPHIC_COLOR layer, OpenCard renders the text data using a TrueType font of the specified font-family at the specified font-size at the specified [x,y] location on the card. Other text attributes may be defined.

When used in a MAGSTRIPE layer, the personalization data is simply be encoded to the magnetic stripe track corresponding to the "datacard:trackType" defined.

Refer to the "Text Element Layer Example" on page 42 for a practical example of how text ID layers are constructed.

There are two kinds of IDs that are used in OpenCard systems: dynamic text IDs and static text IDs.

Dynamic Text Elements

Dynamic text describes an element whose personalization is to be taken in whole or in part from an OpenCard data stream. A dynamic text ID must be of the form "LINE*n*" where *n* can be from 1 to 15 (for example, “LINE1”, “LINE2”, …, “LINE15”). An ID of “LINE1” uses data supplied in the first personalization data line of an OpenCard data stream.

Static Text Elements

If a text element is static, non-changing personalization, define it using these rules:

- Use any ID value other than the form “LINE*n*”.
- Set the attribute datacard:staticElement to “true” to signal that this text item must not be dropped due to there being no ID match with any OpenCard data stream line data.
- Add the static data content at the end of the element.
- Include the end-of-text element (</text>) after the last static data character.

x Attribute

The x attribute defines the distance from the left edge of a card to the left edge of the first character of the text line. Default units are in pixels (px).

y Attribute

The y attribute defines the distance from the top edge of a card to the baseline of the text line. Default units are in pixels (px).

Font-Family Attribute

The font-family attribute selects the font for rendering this text line. To view a list of the font families that are currently installed in the printer, in Printer Dashboard select Main Menu ☰ **> Personalization Tools > OpenCard Configuration**.

Font-Size Attribute

The font-size attribute sets the size at which the text is rendered. Default units are pixels (px), but point size (pt) also is accepted.

At 300 dpi:

pixel size = (point size / 72) * 300

OR

pixel size = point size * 4.167

Font-Weight Attribute

The font-weight attribute specifies the rendering weight of text characters as “normal” or “bold”. The default is “normal” if the attribute is not defined.

Fill Attribute

The fill attribute defines the text color. Find the list of color keyword names (for example, fill=”blue”) at:

http://www.w3.org/TR/SVG11/types.html#ColorKeywords

Color has full effect when used in a GRAPHIC_COLOR personalization operation, but also has an effect on intensity in a GRAPHIC_MONOCHROME operation even though the hue is not printed.

Transform Attribute (Rotating Text)

The transform attribute is an SVG means of performing operations on the entire text element. The most likely used transformation for card format designers is rotating a text element. SVG allows rotation to any degree.

```
x=”100” y=”400” transform=”rotate(90 100, 400)”
```

datacard:staticElement Attribute

When data for the text element is static, this attribute must be set to “true”. The static data content must be added at the end of the element and the end-of-text element (</text>) must be included after the last static data character.

Valid values are “true” and “false”. The default is “false” if this attribute is not defined.

datacard:staticElement Example

```
<g><text id="NameHeader" fill="black" x="75" y="300" font-size="12pt" font-weight="bold"
       font-family="DejaVu Serif” datacard:staticElement="true">Name:</text></g>
```

datacard:appendData Attribute

When the first part of the personalization data for the text element is static and the remainder comes from the OpenCard data stream, this attribute must be set to “true”. The static data content part must be added at the end of the element and the end-of-text element (</text>) must be included after the last static data character.

Valid values are “true” and “false”. The default is “false” if this attribute is not defined.

datacard:appendData Example

```
<g><text id="LINE3" fill="black" x="75" y="525" font-size="11pt" font-family="DejaVu Serif"
datacard:appendData="true">Expires </text></g>
```

datacard:format Attribute

Use this attribute to:

- Perform a character-by-character type validation.
- Insert data into the dynamic personalization data.
- Use only the first "*n*" characters from the personalization data.

The following type checking characters may be used to validate characters in the data:

Checking	Definition
9	Numeric only {0 .. 9}
A	Alphabet only {A .. Z, a .. z}
N	Alphabet or numeric {0 .. 9, A .. Z, a .. z}
X	Any character acceptable

datacard:format Example #1

To render a fixed-length bar code text item such as EAN8, the data item from this sample data stream supplies 7 numeric characters. In this case, the following datacard:format attribute can be defined:

```
datacard:format=”9999999”
```

- If the data stream sends “1234567”, then this passes the type checking defined by the datacard:format attribute. The EAN8 bar code is rendered successfully.
- If the data stream sends “1234567890123”, then the first 7 characters still successfully pass the type checking. The EAN8 bar code is rendered successfully using only the first seven characters “1234567”.
- If the data stream sends “1234A67”, then the data fails the type checking with an error “Format requires numeric character”. The card is rejected.

datacard:format Example #2

Use the datacard:format attribute to insert characters when:

- The data stream does not contain all of the data characters that must print on the card.
- The characters missing from the data stream are placed in the same place in the final data.

For example, use the datacard:format attribute if the data stream supplies an expiration date as “1016” but the printed card must use the date formatted as “10/16”. In this case, use the following datacard:format attribute:

```
datacard:format=”99/99”
```

This inserts the ‘/’ character and verifies that the four data stream characters are numeric.

datacard:remove Attribute

The values accepted are whole number representations only. The default is zero.

If a datacard:remove attribute is defined, then the value of this attribute is used to cut that many data characters from the front of the data stream item that corresponds with the element’s ID.

datacard:remove Example

OpenCard data stream	<1234567890>
Card format	<g><text id="LINE1" fill="black" x="375" y="300" datacard:remove=”3” font-size="12pt"font-family="DejaVu Serif"/></g>
Data merged into card format	<g><text id="LINE1" fill="black" x="375" y="300" datacard:remove=”3” font-size="12pt"font-family="DejaVu Serif">4567890</text></g>

Text Element Layer Example

Text Element Layer Example

This example shows a GRAPHIC_MONOCHROME layer containing text element layers. The data merged into the card format becomes the instructions for rendering the monochrome text and graphics data.

OpenCard data stream

```
<John Doe
12345
Dec. 31, 2012>
```

Card format

```
<g id="GRAPHIC_MONOCHROME">

       <g><text id="NameHeader" fill="black" x="75" y="300"font-size="12pt"
                  font-weight="bold"font-family="DejaVuSerif”
                  datacard:staticElement="true">Name:</text>
       </g>


       <g><text id="LINE1" fill="black" x="375" y="300"font-size="12pt"
                  font-family="DejaVu Serif"/>
       </g>


       <g><text id="PlayerIdHeader" fill="black" x="75" y="400"font-
       size="12pt"
                  font-weight="bold"font-family="DejaVu Serif"
                  datacard:staticElement="true">Player ID:</text>
       </g>


       <g><text id="LINE2" fill="black x="375" y="400"font-size="12pt"
                  font-family="DejaVu Serif" datacard:format="XXXXX"/>
       </g>


       <g><text id="LINE3" fill="black x="75" y="525"font-size="11pt"
                  font-family="DejaVu Serif"
                  datacard:appendData="true">Expires </text>
       </g>
</g>
```

Data merged into card format

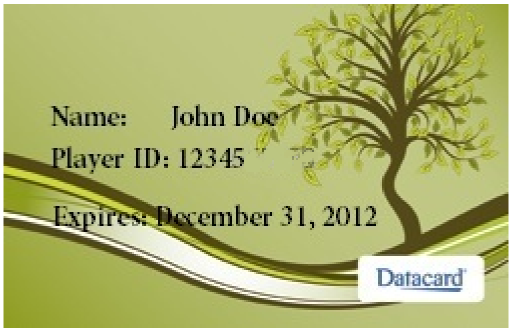
```
<g id="GRAPHIC_MONOCHROME">

       <g><text id="NameHeader" fill="black" x="75" y="300"font-size="12pt"
                  font-weight="bold" font-family="DejaVu Serif”
                  datacard:staticElement="true">Name:</text>
       </g>

       <g><text id="LINE1" fill="black" x="375" y="300"font-size="12pt"
                  font-family="DejaVu Serif"/>John Doe</text>
       </g>

       <g><text id="PlayerIdHeader" fill="black" x="75" y="400"font-
       size="12pt"
                  font-weight="bold" font-family="DejaVu Serif"
                  datacard:staticElement="true">Player ID:</text>
       </g>

       <g><text id="LINE2" fill="black" x="375" y="400" font-size="12pt"
                  font-family="DejaVu Serif"
                  datacard:format="XXXXX"/>12345</text>
       </g>


       <g><text id="LINE3" fill="black" x="75" y="525" font-size="11pt"
                  font-family="DejaVu Serif"
                  datacard:appendData="true">Expires Dec. 31, 2012</text>
       </g>
</g>
```

Card result

Static text IDs

Text id **NameHeader** with the static text “**Name**”

Text id **PlayerIdHeader** with the static text “**Player ID**”



Dynamic text IDs

Text id **LINE1** dynamically pulls **John Doe** from line 1 of the data stream.

Text id **LINE2** dynamically pulls **12345** from line 2 of the data stream.

Using Bar Code-Specific Attributes

Datacard software has built-in support for the following one-dimensional bar code symbologies:

- CODE39
- CODE128
- Interleaved 2 of 5
- UPCA
- EAN8
- EAN13

These may be called out in any GRAPHIC_MONOCHROME or GRAPHIC_COLOR personalization operation using a text layer. Because the bar code patterns for these built-in bar codes are not generated from a TrueType font, no TrueType font needs to be installed to the printer. However, if human-readable characters are required, the Datacard TrueType font “DCP OCR-B.ttf” must be installed in the printer.

The following attributes apply to the Datacard built-in bar codes. The designer is free to install any other bar code as a standard TrueType font and use standard text layer attributes to work with it.

datacard:barcode Attribute

Valid values are “true” and “false”. The default is “false” if the attribute is not defined.

If “true”, this attribute causes the software to interpret the “font-family” attribute value to be a built-in bar code selection as follows. All font-family names are case sensitive and must be exactly as specified:

Bar code symbology	Font-family
CODE39	font-family=”Code39”
CODE128	font-family=”Code128”
Interleaved 2 of 5	font-family=”I2Of5”
UPCA	font-family=”UPC-A”
EAN8	font-family=”EAN-8”
EAN13	font-family=”EAN-13”

datacard:barRatio Attribute

This attribute specifies the ratio of thickness between a narrow bar and a wide bar for a Code39 bar code. This attribute has no effect on any other built-in bar code. The values supported are:

- datacard:barRatio="2to1" (default if no attribute present)
- datacard:barRatio="3to1"

The final narrow and wide bar thickness for Code39 is determined by a combination of the datacard:barRatio and datacard:barDensity values as the datacard:barDensity attribute sets the narrow bar width.

datacard:barDensity Attribute

This attribute effectively sets the narrow bar width for bar codes CODE39, CODE128 and Interleaved 2 of 5 only and has no effect on the generation of bar codes UPCA, EAN8 and EAN13.

Bar code symbology	**Density value**	**Narrow bar width**
CODE39	datacard:barDensity=”4.6”	4
	datacard:barDensity=”5.76”	4
	datacard:barDensity=”6.25”	3
	datacard:barDensity=”7.69”	3
CODE128	datacard:barDensity=”narrow”	3
	datacard:barDensity=”wide”	4
Interleaved 2 of 5	datacard:barDensity=”narrow”	2
	datacard:barDensity=”medium”	3
	datacard:barDensity=”wide”	4
	datacard:barDensity=”extrawide”	5

datacard:barHumanReadable Attribute

Valid values are “true” and “false”. The default is “false” if the attribute is not defined.

If “true”, the bar code is printed with human readable characters if the bar code symbology supports human readable characters. Interleaved 2 of 5 and Code128 do not support human readable characters.

datacard:barChecksum Attribute

Valid values are “true” and “false”. The default is “false” if the attribute is not defined.

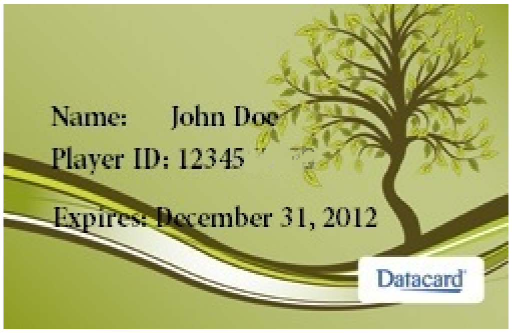
If “true”, Datacard software generates a checksum using a checksum generation algorithm associated with the selected bar code symbology according to this table:

Bar code symbology	Check digit algorithm
CODE39	Modulo 43
CODE128	Modulo 103
Interleaved 2 of 5	Modulo 10
UPCA	Modulo 10
EAN8	Modulo 10
EAN13	Modulo 10

Defining Image Element Layers

Images are supported as static only. This means that data stream items may not be used in conjunction with images.

Images are printed bottom-to-top in the order that they appear in the card format. For example, if the full card tree graphic is followed by the Datacard logo, the tree graphic prints on the bottom with the Datacard logo printed on top.



There is no limit to the number of images that can be rendered to a personalization operation. However, card production throughput may be adversely affected by use of increasing quantities of images, as it takes time to read and process images in the printer.

Defining Image ID Names

For image IDs, choose any name that is useful to the card format, other than the OpenCard data naming pattern “LINE*n*”, which is reserved for dynamic text data.

datacard:positionReference Attribute

Valid values are:

- topLeft (SVG standard compatible option and default)
- bottomLeft

If “topLeft” is specified, then the image y attribute will be a measure from the top edge of a card to the top edge of the image.

If “bottomLeft” is specified, then the image ‘y’ attribute will be a measure from the bottom edge of a card to the bottom edge of the image.

x Attribute

The x attribute defines the distance from the left edge of a card to the left edge of the image. Default units are in pixels (px).

y Attribute

Refer to “datacard:positionReference Attribute” on page 47.

Default units are in pixels (px).

Height Attribute

The height attribute defines the height that the image is scaled to when printed on the card. Units default to pixels (“px”). If no height is defined, then the image is rendered vertically pixel-for-pixel at 300 dpi. If a height is defined, then the image is scaled to the height specified, no matter how many pixels are defined in the image.

Width Attribute

The width attribute describes the width that the image should be scaled to when printed on the card. Units default to pixels (px). If no width is defined, then the image is rendered horizontally pixel-for-pixel at 300 dpi. If a width is defined, then the image is scaled to the width specified, no matter how many pixels are defined in the image.

Transform Attribute (Rotating Images)

The transform attribute is an SVG method of performing operations on the entire image element. SVG allows rotation to any degree.

transform Attribute Example

```
x=”100” y=”200” transform=”rotate(90 100, 200)”
```

Defining the Image Name with xlink:href

Use the xlink:href element to define the case-sensitive name of an image. Load images referenced in a card format onto the printer using Printer Dashboard prior to printing a card with that card format. Image file path information is not required, and is disregarded at production time.

Image Element Example

The following example places the image “TreeDebit.jpg” at the upper left corner of the card (x=0 and y=0), and scales it to the full height and width of a card at 300 dpi (height=638 and width=1013).

```
<g><image id="Background" datacard:staticElement="true" y="0" x="0"height="638px"
      width="1013px" xlink:href="TreeDebit.jpg" /></g>
```

Chapter 4: OpenCard Commands

4

This chapter provides information on using OpenCard commands.

A data stream sent from a host computer might contain some or all of the commands listed in the following table. You normally do not make any changes to OpenCard commands in the data stream. However, it is helpful to know the results of using these commands. This chapter describes how the printer responds to the following OpenCard commands.

Command Description	**Control Codes**
Start of Card Data	<
End of Card Data	>
Graphics Template (change active card layout)	@Gx
Card Stock	@Cxxxx
Magnetic Stripe Encoding	“
New Line	CR or CR-LF or LF-CR or LF

When the printer has an active card layout with one or more fields defined, it ignores all OpenCard commands except the following: Start of Card Data, End of Card Data, and Graphics Template

Start of Card Data Command

The Start of Card Data command is a required command that signals the beginning of printable data to the printer.

Control Code

```
< or STX
```

Example

This data stream shows the Start of Card Data and End of Card Data commands:

```
<
123456
>
```

Comments

The hex codes for the Start of Card Data command are 3C or 02.

End of Card Data Command

The End of Data command is a required command that tells the printer that the card data is complete.

Control Code

```
> or ETX
```

Example

This data stream shows the Start of Card Data and End of Card Data commands:

```
<
123456
>
```

Comments

If you send a data stream that does not include the End of Card Data command, the printer waits 20 seconds and then clears the job.

The hex codes for the End of Data command are 3E or 03.

Graphics Template Command

The Graphics Template command specifies which card layout to use for data streams sent to the printer.

Control Code

```
@Gx
```

where:

@G	Control code
x	The file name of the card layout to use

Example

In this example, the command instructs the printer to use CardLayoutFile.svg for the first card. Then, for the second card, the command instructs the printer to use CardLayoutFile2.svg for the second card.

```
<1222
Preston E. Olson
“%PRESTON OLSON;?1234567890?_;1222?
@GCardLayoutFile.svg>
<3444
Christopher L. Carlson
“%CHRISTOPHER CARLSON;?0987654321?_;3444?
@GCardLayoutFile2.svg
>
```

Notes

The @G command stays in effect until the next @G command, or until the printer is powered off.

The printer displays an error if the card layout filename specified in the command does not exist.

With OpenCardLegacyMode enabled, the printer requires card layout file names of 0, 1, 2, or 3 only. The number corresponds to the allowed legacy layout numbers (0-3). The printer treats the text on the same line after the number as line data. If you send this command with a card layout number outside the allowed range (for example, G6), the printer LCD panel displays error message, 100: Request not supported. Refer to the printer’s User’s Guide for more information.

If an @G command appears in the data stream following magnetic stripe data, subsequent data in the data stream is extracted and printed. This provides support for data streams for legacy printers which embossed, encoded magnetic stripe data, and then printed on the card. It also allows you to add, using the Card Layout menus, one or more logos or fixed text fields.

Card Stock Command

The @C command in the OpenCard data stream defines which card stock to use. A card stock definition is required when setting up the LCD panel to prompt the user to insert a card in the exception slot of the input hopper. Prompts from the LCD panel are shown only with manual card insertion.

Control Code

@Cxxxx

Where:

@C = Control code

xxxx = The name of the card stock as specified in Printer Dashboard.

Example

```
<1222
Preston E. Olson
“%PRESTON OLSON;?1234567890?_;1222?
@GCardLayoutFile.svg
@CCardStock1>
<3444
Christopher L. Carlson
“%CHRISTOPHER CARLSON;?0987654321?_;3444?
@GCardLayoutFile2.svg
@CCardStock2
>
```

Notes

- If an @C command is in the data stream, then the printer uses the card stock requested.
- If the card stock is not defined in the data stream, then the printer uses the card stock defined in Printer Dashboard as **Default**. If **Default** doesn’t exist, the card is rejected and the request does not print.
- If you are using logical hopper groups, ensure that the name used here matches the name of the logical hopper group configured in Printer Dashboard. For more information on logical hopper groups, refer to the Printer Dashboard online Help.

Magnetic Stripe Encoding Data Command

For printers with a 3-track magnetic stripe module, this command tells the printer to encode the data following the control code on the card's magnetic stripe.

Control Code

"

Notes

Magnetic stripe data can be included within the data delimited by the Start of Card Data and End of Card Data commands (< and >).

The data is encoded according the track formats set in the printer. The printer is shipped with the default track formats, depending on the magnetic stripe module. The following table summarizes requirements for each default track.

Track	Default Encoding Format	Start Sentinel	End Sentinel	Data Allowed	Maximum Characters
1	IATA—International Air Transportation Association	% (25 hex)	? (3F hex)	Capital letters, numbers, a space, and ! # $ % ‘ ( ) * + , - . / ; : < @ > = ^] \ [" & _	76
2	ABA—American Bankers Association	; (3B hex)	? (3F hex)	Numbers and ; : < > =	37
3	TTS—Thrift Third Shift	_ (5F hex) or _; (5F3B hex)	? (3F hex)	Numbers and ; : < > +	104

The data must contain the start and end sentinels for each track and cannot include carriage returns or line feeds. The start and end sentinels mark the beginning and end of each track. The host computer must send data that meets the requirements for each track.

Refer to your printer’s *Installation and Administrator’s Guide* for more information about magnetic stripe encoding.

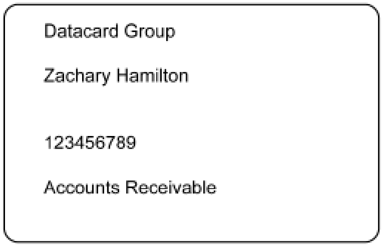
Example

In this example, the text on lines 1, 3, 6, and 8 will be printed on the front of the card. The employee number (123-456-789) will be encoded on track 2 of the magnetic stripe and the access code (4321) will be encoded on track 3 of the magnetic stripe. All of the information is applied to the card as part of one print job.

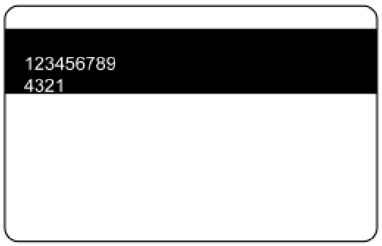
```
<Datacard GroupLF
LF
Zachary HamiltonLF
LF
LF
123-456-789LF
LF
Accounts ReceivableLF
LF
";123456789?_;4321?LF
>
```

The following shows the card produced by sending the example data stream to the printer:

Front of Card



Back of Card—Encoded on Magnetic Stripe



Track 1—no data
Track 2—Employee Number
Track 3—Access Code

Note: Data encoded on the magnetic stripe is not visible.

Comments

The hex code for the Magnetic Stripe Encoding Data command is 22.

New Line Command

The New Line command tells the printer to start a new line of printed data.

Control Code

```
CR | CR-LF | LF-CR | LF
```

Description

When a print line is sent to the printer, the computer system normally inserts one of the four New Line commands (line feed or carriage return) at the end of each line. These codes trigger the printer to start a new line on the card according to the default line locations. If a data stream contains blank lines, the printer skips those lines and moves down the card.

Example

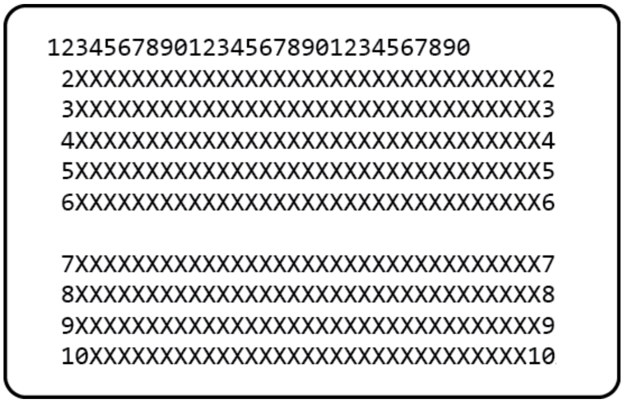
In this example, the data stream causes lines of X characters to be printed on the card, and skips the blank line between lines 6 and 7. This is how the example data stream appears in a text file:

```
<123456789012345678901234567890
 2XXXXXXXXXXXXXXXXXXXXXXXXXXX2
 3XXXXXXXXXXXXXXXXXXXXXXXXXXX3
 4XXXXXXXXXXXXXXXXXXXXXXXXXXX4
 5XXXXXXXXXXXXXXXXXXXXXXXXXXX5
 6XXXXXXXXXXXXXXXXXXXXXXXXXXX6

 7XXXXXXXXXXXXXXXXXXXXXXXXXXX7
 8XXXXXXXXXXXXXXXXXXXXXXXXXXX8
 9XXXXXXXXXXXXXXXXXXXXXXXXXXX9
 10XXXXXXXXXXXXXXXXXXXXXXXXX10
>
```

This is how the printer reads the same data stream. The line feed characters are represented below as `LF`.

```
<123456789012345678901234567890LF
 2XXXXXXXXXXXXXXXXXXXXXXXXXXX2LF
 3XXXXXXXXXXXXXXXXXXXXXXXXXXX3LF
 4XXXXXXXXXXXXXXXXXXXXXXXXXXX4LF
 5XXXXXXXXXXXXXXXXXXXXXXXXXXX5LF
 6XXXXXXXXXXXXXXXXXXXXXXXXXXX6LF
LF
 7XXXXXXXXXXXXXXXXXXXXXXXXXXX7LF
 8XXXXXXXXXXXXXXXXXXXXXXXXXXX8LF
 9XXXXXXXXXXXXXXXXXXXXXXXXXXX9LF
 10XXXXXXXXXXXXXXXXXXXXXXXXX10LF
>
```

The following shows the card produced by sending the example data stream to the printer:



```
123456789012345678901234567890
 2XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX2
 3XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX3
 4XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX4
 5XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX5
 6XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX6

 7XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX7
 8XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX8
 9XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX9
 10XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX10
```

Comments

Hex codes for the New Line commands are as follows:

CR = 0D

LF = 0A

CR-LF = 0D and 0A

LF-CR = 0A and 0D

Chapter 5: Exporting Legacy Card Layouts, Graphics, and Fonts

5

This chapter provides information about using existing Legacy card layouts from SP/CP Series printers to print on Sigma Series printers with OpenCard. This includes how to export Legacy card layouts, how to use Legacy graphics/logos, and how to use Legacy fonts.

Exporting Card Layouts from Legacy Series Printers with Telnet

A card layout refers to Legacy card layout files originally created for the SP/CP Series card printers. You can export these card layouts, save the exported definition to a file, and then import the card layout files for use with OpenCard on Sigma Seriesprinters.

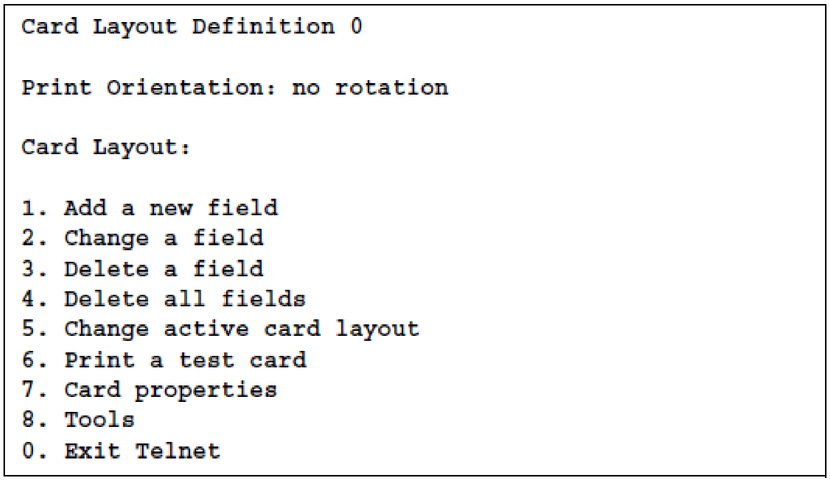
For more information about setting up Legacy SP/CP Series card layouts, refer to the Legacy SP/CP Series *Data Formatting Guide*.

1. Make sure that the PC is connected to the network using TCP/IP.
2. Establish a Telnet connection to the printer.
3. Enter the following at the command prompt or command line:

   ```
   telnet ipaddress
   ```

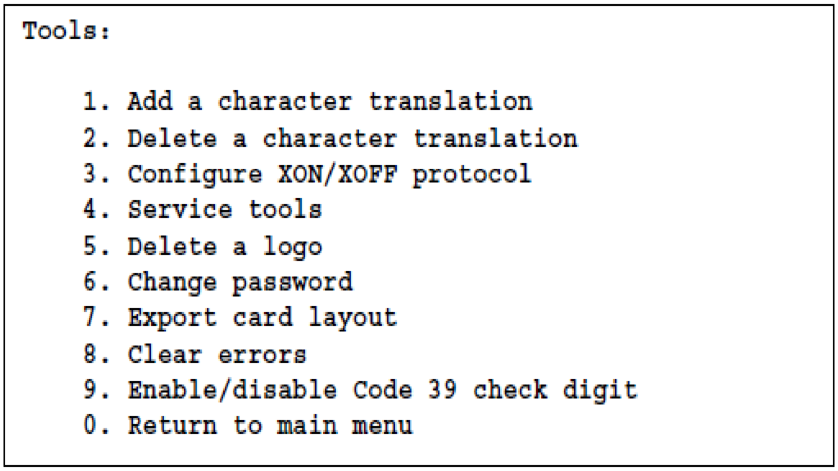
   Where *ipaddress* is the IP address displayed on the LCD panel of the printer (**Printer Ready>Main Menu>Status>Status Menu>Network>Address Mode>DHCP (or Static)>Data Format>Open Card>IP Address>IP address of printer**).

The Card Layout Definition menu displays.



```
Card Layout Definition 0

Print Orientation: no rotation

Card Layout:

1. Add a new field
2. Change a field
3. Delete a field
4. Delete all fields
5. Change active card layout
6. Print a test card
7. Card properties
8. Tools
0. Exit Telnet
```

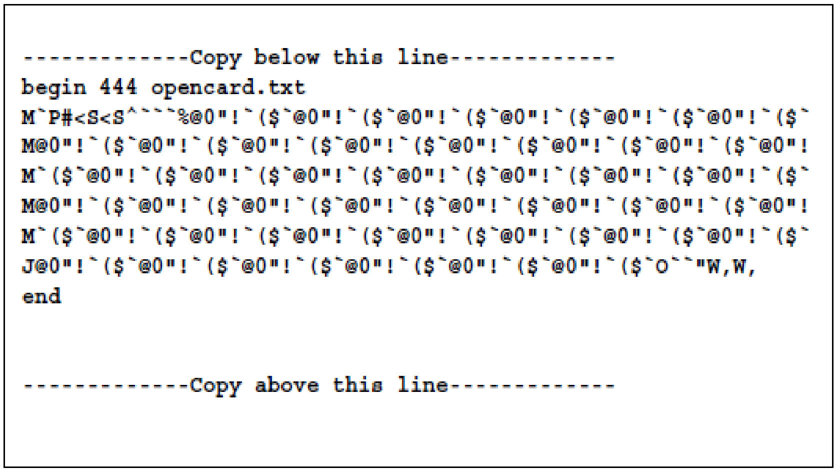
4. Make sure that the desired active card layout is displayed. Only the current active card layout and any currently defined character translations is exported.
5. From the Card Layout Definition menu, enter **8** to select **Tools**. The Tools menu displays.



```
Tools:

    1. Add a character translation
    2. Delete a character translation
    3. Configure XON/XOFF protocol
    4. Service tools
    5. Delete a logo
    6. Change password
    7. Export card layout
    8. Clear errors
    9. Enable/disable Code 39 check digit
    0. Return to main menu
```

6. From the Tools menu, enter **7** to select **Export card layout**.

The printer displays the card layout, followed by the Tools menu. The beginning and end of the card layout are marked with the words “begin” and “end.” The card layout is encoded.

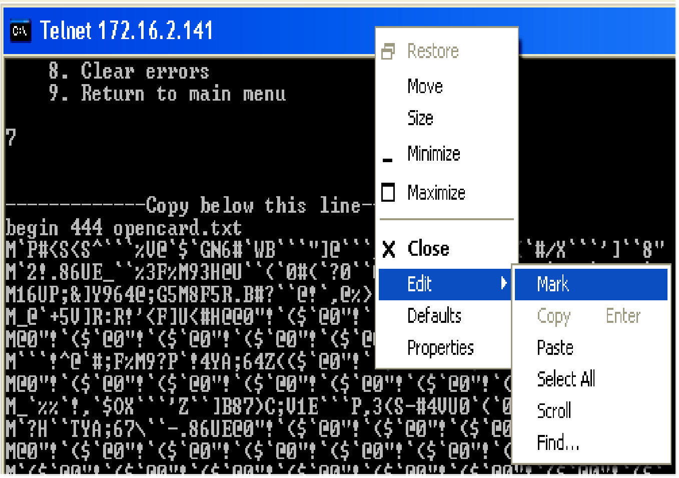


```
-------------Copy below this line-------------
begin 444 opencard.txt
M`P#<S<S^```%@0"!`($`@0"!`($`@0"!`($`@0"!`($`@0"!`($`@0"!`($`
M@0"!`($`@0"!`($`@0"!`($`@0"!`($`@0"!`($`@0"!`($`@0"!`($`@0"!
M`($`@0"!`($`@0"!`($`@0"!`($`@0"!`($`@0"!`($`@0"!`($`@0"!`($`
M@0"!`($`@0"!`($`@0"!`($`@0"!`($`@0"!`($`@0"!`($`@0"!`($`@0"!
M`($`@0"!`($`@0"!`($`@0"!`($`@0"!`($`@0"!`($`@0"!`($`@0"!`($`
J@0"!`($`@0"!`($`@0"!`($`@0"!`($`@0"!`($`@0"!`($`O``"W,W,
end


-------------Copy above this line-------------
```

7. Use the tools available in your Telnet interface to copy the exported card layouts, and then save the layouts in a text file.

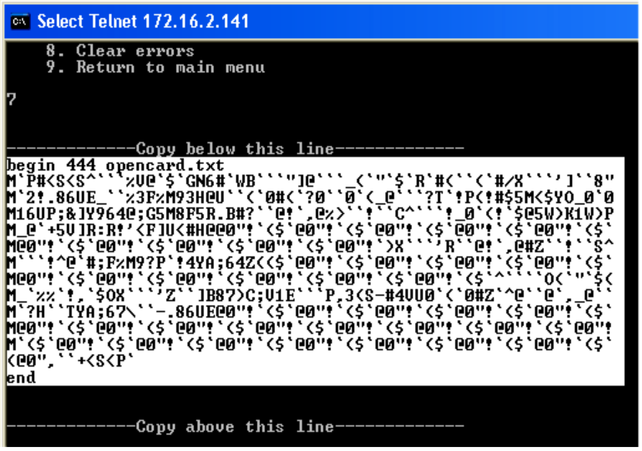
   For example, if you are using Telnet with the Windows command prompt, do the following:

   a. Right-click the title bar to display a pop up menu. Select **Edit > Mark** from the menu.



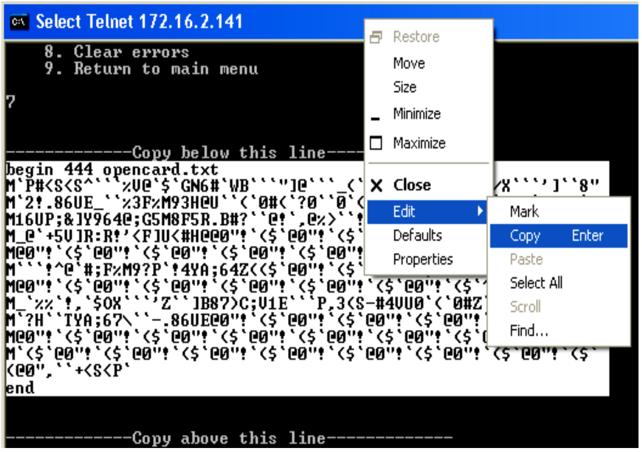
```
    8. Clear errors
    9. Return to main menu

7


-------------Copy below this line-
begin 444 opencard.txt
```

b. Highlight the layout. Start at “Begin 444 opencard.txt” and go to “end.”



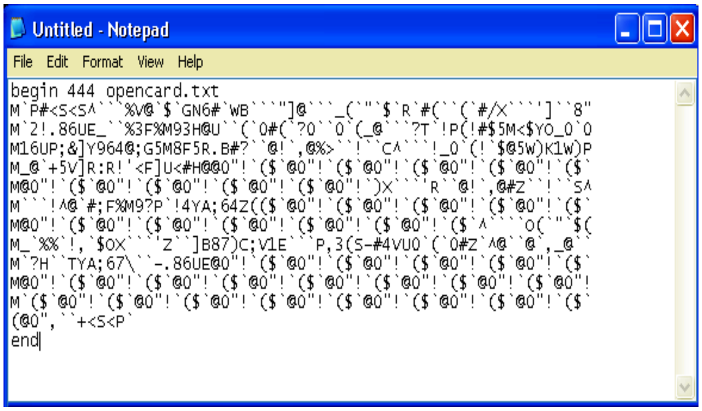
```
     8. Clear errors
     9. Return to main menu

7

-------------Copy below this line-------------
begin 444 opencard.txt
M`P#<S<S^```%V@`$`GN6#`WB```"]@```_(`"`$`R`#(``(`#/X```']``8"
M`2!.86UE_``%3F%M93H@U``(`0#(`?0``0`(_@```?T`!P(!#$5M<$YO_0`0
M16UP;&]Y964@;G5M8F5R.B#?``@!`,@%>``!``C^```!_0`(!`$@5W)K1W)P
M_@`+5V]R:R!'<F]U<#H@@0"!`($`@0"!`($`@0"!`($`@0"!`($`
M@0"!`($`@0"!`($`@0"!`($`@0"!`)X```'R``@!`,@#Z``!``S^
M```!^@`#;F%M9?P`!4YA;64Z(($`@0"!`($`@0"!`($`@0"!`($`
M@0"!`($`@0"!`($`@0"!`($`@0"!`($`^````O(`"`$(
M_`%%`!,`$OX```'Z``]B87)C;V1E```P,3(S-#4VU0`(`0#Z`^@``@`,_@``
M`?H``TYA;67\``-.86UE@0"!`($`@0"!`($`@0"!`($`@0"!`($`
M@0"!`($`@0"!`($`@0"!`($`@0"!`($`@0"!`($`@0"!
M`($`@0"!`($`@0"!`($`@0"!`($`@0"!`($`@0"!`($`@0"!`($`
(@0",``+<S<P`
end

-------------Copy above this line-------------
```

c. Right-click the title bar and select **Edit > Copy** from the pop-up menu.



d. Open Notepad or a similar text editing application.

e. Paste the copied text in the text editing application.



```
begin 444 opencard.txt
M`P#<S<S^```%V@`$`GN6#`WB```"]@```_(`"`$`R`#(``(`#/X```']``8"
M`2!.86UE_``%3F%M93H@U``(`0#(`?0``0`(_@```?T`!P(!#$5M<$YO_0`0
M16UP;&]Y964@;G5M8F5R.B#?``@!`,@%>``!``C^```!_0`(!`$@5W)K1W)P
M_@`+5V]R:R!'<F]U<#H@@0"!`($`@0"!`($`@0"!`($`@0"!`($`
M@0"!`($`@0"!`($`@0"!`($`@0"!`)X```'R``@!`,@#Z``!``S^
M```!^@`#;F%M9?P`!4YA;64Z(($`@0"!`($`@0"!`($`@0"!`($`
M@0"!`($`@0"!`($`@0"!`($`@0"!`($`^````O(`"`$(
M_`%%`!,`$OX```'Z``]B87)C;V1E```P,3(S-#4VU0`(`0#Z`^@``@`,_@``
M`?H``TYA;67\``-.86UE@0"!`($`@0"!`($`@0"!`($`@0"!`($`
M@0"!`($`@0"!`($`@0"!`($`@0"!`($`@0"!`($`@0"!
M`($`@0"!`($`@0"!`($`@0"!`($`@0"!`($`@0"!`($`@0"!`($`
(@0",``+<S<P`
end
```

f. Save the file using the following naming conventions:

- If you want this card layout to be the default SP/CP Series card format, then save to the file named Default.

  The file name Default is case sensitive. Make sure to save the file as Default with no extension.

- If you use the @G command to indicate the name of the card layout in the data stream, you must save the file using one of the following names:
  - 0
  - 1
  - 2
  - 3
- Where 0, 1, 2, or 3 corresponds to the specified @G0, @G1, @G2 or @G3 name of the card layout in the data stream. Make sure to save the file with no extension.

8. Import card formats using Printer Dashboard. For instructions, refer to the Printer Dashboard online Help.

Using Legacy Graphic/Logo Files

Legacy graphic/logo files are not exported automatically with the card layout. Instead, you must import legacy graphics separately using Printer Dashboard. For more information about importing Legacy files using Printer Dashboard, refer to the Printer Dashboard online Help.

When importing legacy logos originally created for SP/CP Series printers card layouts, you must use the original monochrome logos in TIFF format and name them one of the following:

- OpencardLogo1
- OpencardLogo2
- OpencardLogo3
- OpencardLogo4

Where 1, 2, 3, and 4 correspond to the original graphic/logo name in the data stream.

Using Legacy Fonts

To print to the Sigma Series printers, Legacy fonts are bundled with the printer. OpenCard maps Legacy SP/CP Series printer fonts to their TrueType equivalents resident on the Sigma Series printers, as shown below.

SP/CP Series Card Printers Equivalent	Sigma Series Printers with OpenCard
Serif	ufonts.com_charter-bt-roman.ttf
Serif Bold	ufonts.com_charter-bt-black.ttf
Courier	ufonts.com_courier-10-pitch-bt.ttf
Courier Bold	ufonts.com_courier-10-pitch-bold-bt.ttf
Sans	ufonts.com_bitstream-vera-sans.ttf
Sans Bold	ufonts.com_bitstream-vera-sans-bold.ttf
	Used for bar code human-readable characters: DCP OCR-B.ttf

Some legacy printers use custom fonts designed specifically for that system. You must manually load any custom fonts onto the printer.

Chapter 6: Working with Printer Dashboard

Printer Dashboard is a web interface that provides tools for configuring images, fonts, card stocks, and card formats. This chapter describes how to import card formats, images, and fonts, how to define card stocks, and how to view the Print Request Log and the OpenCard data stream.

Prepare for OpenCard Printing

Before a printer can print using OpenCard you must upload resources to the printer. Complete the following tasks to prepare the printer for OpenCard printing:

1. Enable OpenCard Personalization Tools on page 66
2. Add a Card Format on page 66
3. Add a Card Stock Definition on page 67
4. Add Images on page 67
5. Add Fonts on page 67

Open Printer Dashboard

Open Printer Dashboard to configure OpenCard on the printer.

1. Use the printer's LCD menu to get the printer IP address. Refer to the printer's User's Guide for information.
2. Open a web browser on the computer then enter the following address:

   https://*[printer IP address]*/

   The browser displays a certificate warning because it is a secure connection.
3. Click **Continue to this website (not recommended)**. The Log In page displays.
4. Enter a **User ID** and **Password** then click **Login**. The Printer Dashboard opens.

Enable OpenCard Personalization Tools

OpenCard Personalization Tools in Printer Dashboard include tools to upload SVG images for

if it is not enabled - if they cannot see the Personalization Tools option in the menu, enable

Contact Entrust service if the printer is not enabled for OpenCard.

1. In Printer Dashboard, select Main Menu ☰ **> Configuration > Settings**. The Settings page opens.
2. From the drop-down list below Change Settings, select **Behavior**.
3. Set the value for the **Plugin** setting to **Enabled**.
4. Click **Save**.
5. Restart the printer.
    a. Select Main Menu ☰ **> Troubleshooting > Restart Printer**. The Restart Printer page opens.
    b. Click **Restart**. The printer restarts.

Configure OpenCard

To prepare a printer to print using OpenCard, upload a card format, define a card stock, add all images, and upload fonts.

Add a Card Format

Card formats are an XML-based SVG file that set the design and layout of cards printed using OpenCard. Follow these steps to upload a card format file.

1. In Printer Dashboard, select Main Menu ☰ **> Personalization Tools > OpenCard Configuration**. The OpenCard Configuration page opens.
2. In the **Card Formats** area, click **Add**. The Open dialog box opens.
3. Select a card format file then click **Open**. Printer Dashboard uploads the card format.

Add a Card Stock Definition

Card stock definitions are named in the card format file and found on the printer at production time. In the card stock definition, you specify a card stock name and which images will print on the front and back of the card. A card stock definition is required when setting up the printer LCD panel to prompt the user to insert a card in the exception hopper.

1. In Printer Dashboard, select Main Menu ☰ **> Personalization Tools > OpenCard Configuration**. The OpenCard Configuration page opens.
2. In the **Card Stocks** area, click **Add**. The Add Card Stock dialog box opens.
3. In the **Name** field, enter a name for the card stock.
4. From the **Hopper** list, select the hopper that contains the card stock.
5. From the **Front** list, select an image to print on the front of the card.
6. From the **Back** list, select an image to print on the back of the card.
7. Click **Save**.

Add Images

Upload all images used in the card format or data stream to the printer before printing. Follow these steps to upload images to the printer.

1. In Printer Dashboard, select Main Menu ☰ **> Personalization Tools > OpenCard Configuration**. The OpenCard Configuration page opens.
2. In the **Images** area, click **Add**. The Open dialog box opens.
3. Select and image then click **Open**. Printer Dashboard uploads the image.

Add Fonts

The Fonts area displays all fonts available for the printer to use. If a font you intend to use is not on that list, add the font to the printer.

1. In Printer Dashboard, select Main Menu ☰ **> Personalization Tools > OpenCard Configuration**. The OpenCard Configuration page opens.
2. In the **Fonts** area, click **Add**. The Open dialog box opens.
3. Select a font file then click **Open**. Printer Dashboard adds the font to the printer.

OpenCard Print Log

The OpenCard print log displays all print jobs submitted using OpenCard. Printer Dashboard adds an entry to the log after the printer receives a data stream and print request. The OpenCard print log displays the current status of the print requests.

View OpenCard Print Log

View the OpenCard print log to view a list of all Open Card print jobs submitted to the printer.

In Printer Dashboard, select Main Menu ☰ **> Personalization Tools > OpenCard Print Log**. The OpenCard Print Log page opens.

Reset Print Log

Reset the OpenCard print log to clear inactive print jobs from the print log.

1. In Printer Dashboard, select Main Menu ☰ **> Personalization Tools > OpenCard Print Log**. The OpenCard Print Log page opens.
2. Click **Reset**. A confirmation dialog box opens.
3. Click **Yes**. Printer Dashboard resets the OpenCard print log.

View Data Stream Information

Follow these steps to view information on the most recent OpenCard data stream.

1. In Printer Dashboard, select Main Menu ☰ **> Personalization Tools > OpenCard Print Log**. The OpenCard Print Log page opens.
2. Click **Data Stream**. The Data Stream dialog box opens displaying information on the latest OpenCard data stream.

Export from Printer Dashboard

Export fonts, card formats, or images from the printer to save the files or transfer the files to a different printer.

Export Card Formats

Follow these steps to export card formats from the printer.

1. In Printer Dashboard, select Main Menu ☰ **> Personalization Tools > OpenCard Configuration**. The OpenCard Configuration page opens.
2. Select a card format from the **Card Formats** area.
3. Click **Download**. The browser downloads the card format file.

Export Images

Follow these steps to export images from the printer.

1. In Printer Dashboard, select Main Menu ☰ **> Personalization Tools > OpenCard Configuration**. The OpenCard Configuration page opens.
2. Select an image from the **Images** area.
3. Click **Download**. The browser downloads the image.

Export Fonts

Follow these steps to export font files from the printer.

1. In Printer Dashboard, select Main Menu ☰ **> Personalization Tools > OpenCard Configuration**. The OpenCard Configuration page opens.
2. Select a font file from the **Fonts** area.
3. Click **Download**. The browser downloads the font file.

Chapter 7: Printing Cards

This chapter includes information you need to print cards on Sigma Series printers using OpenCard.

To print cards on Sigma Series printers using OpenCard you must:

- Create a card format. For instructions, refer to "Creating Card Formats" on page 17.
- Import card formats, images, and fonts to the printer, and define card stock using Printer Dashboard. For instructions, refer to "Working with Printer Dashboard" on page 65.
- Have a working data stream.
- Create a card stock. For instructions, refer to "Working with Printer Dashboard" on page 65.

1. Before you start printing, make sure that the printer is at the **Ready** state.
2. Open the Command Prompt (**cmd.exe**) from the Start Menu.
3. Enter `file2prn [filename] ipaddress [portnumber]` where:

   `filename` is the name of a card data stream file.

   `ipaddress` is the IP address of the printer to which you are printing the data stream.

   `portnumber` is the port number at the printer to which you are printing the data stream. If you do not specify a port number, 9100 is assumed.

   The following displays, indicating that the file was sent successfully:

   ```
   sending file.............sent
   ```
4. Repeat steps 2 and 3 for each data stream to be printed.

If a card does not print, view the Print Request Log to find troubleshooting messages about the printer state.

Appendix A: Sample Card Formats

This appendix provides sample card formats with comments. Use it as a reference for creating OpenCard card formats for the Sigma Series printers.

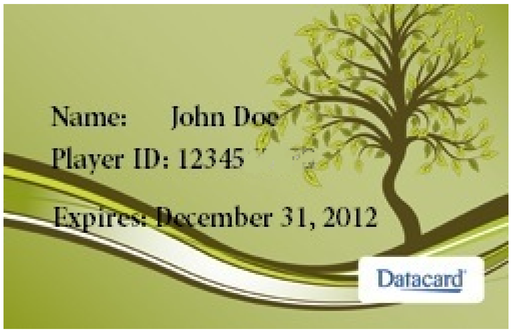
Sample 1—Multipanel Graphics.svg

The card format shown below includes comments that describe how it creates the printed card shown. Note that since the CARD_BACK layer is defined, this card has elements on the back, including color graphics, topcoat, and a magnetic stripe.

Comments describing the sample begin with the <!- - symbols and end with the --> symbols:

This example uses the following data stream:

OpenCard data stream	`<John Doe`
	`1234`
	`December 31, 2012`
	`@GSample1.svg`
	`"%JOHN DOE^0205?;0205:2200000042?;1234567890?>`



```
<?xml version="1.0" encoding="UTF-8"?>
<svg width="1013px" height="638px" xmlns="http://www.w3.org/2000/svg">

   <!-- Front side of card layer. Define color, monochrome and topcoat personalization layers -
->

<g id="CARD_FRONT">

   <!-- Color graphics layer. Scale image "TreeDebit.jpg" to full card size (1013 x 638 pixels)
-->
    <g id="GRAPHIC_COLOR">
       <g><image id="Background" datacard:staticElement="true" y="0" x="0"
                    height="638px" width="1013px" xlink:href="TreeDebit.jpg" /></g>
    </g>

    <!-- Monochrome graphics layer. Two static text items for name and player id headings
printed in bold. Three data stream driven dynamic text items LINE1, LINE2, LINE3 -->
    <!-- LINE2 uses only the first five characters of any type from the data stream (i.e.
datacard:format="XXXXX") -->
    <!-- LINE3 will append the data from data stream LINE3 to the string "Expires ". Note that
this is an alternative to having one static and one dynamic field such as NameHeader + LINE1 -->
    <!-- Note that monochrome always prints "after" (or on top of) YMC color printing regardless
of ordering of layers in card format -->
    <g id="GRAPHIC_MONOCHROME">
      <g><text id="NameHeader" fill="black" x="75" y="300" font-size="12pt" font-weight="bold"
font-family="DejaVu Serif" datacard:staticElement="true">Name:</text></g>
       <g><text id="LINE1" fill="black" x="375" y="300" font-size="12pt" font-family="DejaVu
Serif"/></g>
       <g><text id="PlayerIdHeader" fill="black" x="75" y="400" font-size="12pt" font-
weight="bold" font-family="DejaVu Serif" datacard:staticElement="true">Player ID:</text></g>
       <g><text id="LINE2" fill="black" x="375" y="400" font-size="12pt" font-family="DejaVu
Serif" datacard:format="XXXXX"/></g>
       <g><text id="LINE3" fill="black" x="75" y="525" font-size="11pt" font-family="DejaVu
Serif" datacard:appendData="true">Expires </text></g>
    </g>

   <!-- Topcoat layer.  Scale image "Topcoat_Full.png" which is a 100% black image to full card
size (1013 x 638 pixels) -->
    <!-- Note that topcoat always prints "after" (or on top of) YMC color printing and / or
monochrome printing regardless of ordering of layers in card format -->
    <g id="TOPCOAT">
       <g><image id="TopcoatImage" datacard:staticElement="true" y="0" x="0" height="638px"
width="1013px" xlink:href="Topcoat_Full.png" /></g>
    </g>

  </g>
  <!-- Back side of card layer.  Define color, topcoat and magnetic stripe personalization
layers on this side   -->
  <g id="CARD_BACK">

    <!-- Color graphics layer. Scale image "DatacardNoText.png" to 660 x 150 pixels  -->
    <!-- Position the left edge 170 pixels from the left edge of the card and the top edge 400
pixels from the top edge of the card  -->
    <g id="GRAPHIC_COLOR">
       <g><image id="BackLogo" datacard:staticElement="true" y="400" x="170" height="150"
width="660" xlink:href="DatacardNoText.png" /></g>
    </g>

   <!-- Topcoat layer. Scale image "Topcoat_Full.png" which is a 100% black image to full card
size (1013 x 638 pixels) -->
    <g id="TOPCOAT">
       <g><image id="TopcoatImage" datacard:staticElement="true" y="0" x="0" height="638px"
width="1013px" xlink:href="Topcoat_Full.png" /></g>
    </g>

    <!-- Magnetic stripe layer. -->
    <!-- Use ids of the form "ISOx" to use double-quoted (") magnetic stripe command content
from the data stream or use ids of the form "LINEx" to use text line data from the data stream
-->
    <!-- datacard:trackType must always be one of {"ISO1", "ISO2", "ISO3"} -->
    <g id="MAGSTRIPE">
       <g><text id="ISO1" datacard:trackType="ISO1"/></g>
       <g><text id="ISO2" datacard:trackType="ISO2"/></g>
       <g><text id="ISO3" datacard:trackType="ISO3"/></g>
    </g>

  </g>
</svg>
```

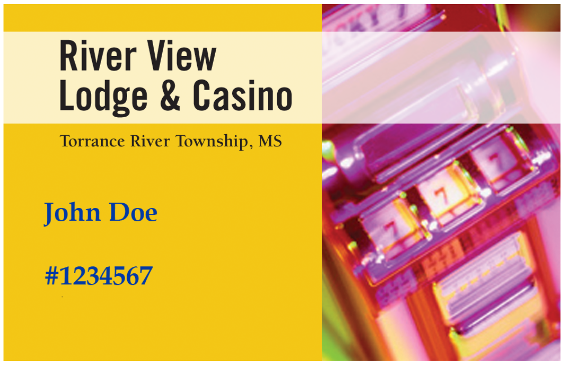
Sample 2—RiverViewCasino.svg

The card format document shown below includes comments as to how it creates the card shown. Note that since the CARD_BACK layer is defined, this card has elements on the back, including a bar code and a magnetic stripe.

Comments describing the sample begin with the <!- - symbols and ended with the --> symbols.

This example uses the following data stream:

OpenCard data stream	`<John Doe`
	`1234567890`
	`@CDefault`
	`@GRiverViewCasino.svg>`



```
<?xml version="1.0" encoding="UTF-8"?>
<svg width="1013px" height="638px" xmlns="http://www.w3.org/2000/svg">

   <!-- Front side of card layer.  Define color, monochrome and topcoat personalization layers
-->
<g id="CARD_FRONT">

    <!-- Color graphics layer.  Scale image "RiverViewCasino.tif" to full card size
         (1013 x 638 pixels) -->
    <!-- Print name using LINE1 data item in blue.  Print ID# using the first seven digits from
         the LINE2 data item in blue, statically prepending a '#' -->
    <g id="GRAPHIC_COLOR">

      <g><image id="Background" datacard:staticElement="true" y="0" x="0" height="638px"
      width="1013px" xlink:href="RiverViewCasino.tif" /></g>

      <g><text id="LINE1" fill="blue" x="75" y="400" font-size="12pt" font-family="DejaVu
      Serif" font-weight="bold"/></g>

      <g><text id="LINE2" fill="blue" x="75" y="525" font-size="12pt" font-family="DejaVu
      Serif" font-weight="bold" datacard:format="#9999999"/></g>
    </g>

    <!-- Topcoat layer.  Scale image "Topcoat_Full.png" which is a 100% black image to full card
size (1013 x 638 pixels) -->
    <!-- Note that topcoat always prints after (or on top of) YMC color printing and / or
monochrome printing regardless of ordering of layers in card format -->
    <g id="TOPCOAT">
       <g><image id="TopcoatImage" datacard:staticElement="true" y="0" x="0" height="638px"
         width="1013px" xlink:href="Topcoat_Full.png" /></g>
    </g>

  </g>

   <!-- Back side of card layer.  Define color, topcoat and magnetic stripe personalization
layers on this side   -->
  <g id="CARD_BACK">


<!-- Monochrome graphics layer.  Print a Code39 barcode using the first seven digits from data
item LINE2.-->
    <!-- Density=4.6 and bar ratio of 3:1, with human readable -->
    <!-- Note that monochrome always prints "after" (or on top of) YMC color printing regardless
of ordering of layers in card format -->
    <g id="GRAPHIC_MONOCHROME">
       <g><text id="LINE2" fill="black" x="225" y="400" font-size="18pt"
      datacard:format="9999999" font-family="Code39" datacard:barcode="true"
      datacard:barHumanReadable="true" datacard:barDensity="4.6" datacard:barRatio="3to1"/></
g>
    </g>
```

```
<!-- Magnetic stripe layer. -->
<!-- Use ids of the form "ISOx" to use double-quoted (") magnetic stripe command content from
the data stream or use ids of the form "LINEx" to use text line data from the data stream -->
    <!-- datacard:trackType must always be one of {"ISO1", "ISO2", "ISO3"} -->
<!-- This example shows how to extract data from LINE2 and use it to encode on ISO1 (Track1).-->
<g id="MAGSTRIPE">
       <g><text id="LINE2" datacard:trackType="ISO1"/></g>
    </g>

  </g>
</svg>
```

Appendix B: Setting Up OS/400

This appendix provides information about how to prepare to use the IBM OS/400 operating system with OpenCard.

The printer supports the LPD/LPR (line printer daemon/line printer remote) protocol to receive OpenCard data from a host computer.

Configuring the OS/400 Operating System

1. To configure the OS/400 operating system, type the following from a command line exactly as shown. This creates the output queue for the printer.

   In this example, the queue name is SPOUTQ. The parameters in italic text are variable.

   ```
   CRTOUTQ OUTQ(QUSRSYS/SPOUTQ) RMTSYS(*INTNETADR)
   RMTPRTQ(RAW) CNNTYPE(*IP) DESTTYPE(*OTHER)
   MFRTYPMDL(*WSCST) WSCST(QSYS/QWPDEFAULT)
   INTNETADR(’XXX.XXX.XXX.XXX’) SEPPAGE(*NO)
   TEXT(’SP Printer’)
   ```

2. To start the printer and start printing, type the following at a command line:

   ```
   STRRMTWTR (QueueName)
   Comments
   ```

3. To stop all printing to the printer, type the following:

   ```
   ENDWTR (QueueName)
   ```

4. To monitor the printer, type the following:

   ```
   WRKOUTQ (QueueName)
   ```

The following shows the result of the WRKOUTQD command:

```
5722SS1 V5R1M0 010525 AS3 03/19/04 11:08:21
Queue: SPOUTQ Library: OLSONP
Status:
Writer active . . . . . . . . . . . . . : Y
Writer name(s) if active . . . . . . . : SPOUTQ
Output queue held . . . . . . . . . . : N
Maximum spooled file size:
Number of pages . . . . . . . . . . . : *NONE
Starting time . . . . . . . . . . . . :
Ending time . . . . . . . . . . . . . :
Writers to autostart . . . . . . . . . : 1
Display any file . . . . . . . . . . . : *NO
Job separators . . . . . . . . . . . . : 0
Operator controlled . . . . . . . . . . : *YES
Order of files on queue . . . . . . . . : *FIFO
Data queue . . . . . . . . . . . . . . : *NONE
Library . . . . . . . . . . . . . . . :
Authority to check . . . . . . . . . . : *OWNER
Remote system . . . . . . . . . . . . . : *INTNETADR
Remote printer queue . . . . . . . . . : RAW
Queue for writer messages . . . . . . . : QSYSOPR
Library . . . . . . . . . . . . . . . : *LIBL
Connection type . . . . . . . . . . . . : *IP
Internet address . . . . . . . . . . . : 111.111.111.111
Destination type . . . . . . . . . . . : *OTHER
VM/MVS class . . . . . . . . . . . . . :
FCB . . . . . . . . . . . . . . . . . . :
Host print transform . . . . . . . . . : *YES
User data transform . . . . . . . . . :
Library . . . . . . . . . . . . . . . :
Manufacturer type and model . . . . . : *WSCST
Workstation customizing object . . . . : QWPDEFAULT
Library . . . . . . . . . . . . . . . . : QSYS
Image configuration . . . . . . . . . . : *NONE
Destination options . . . . . . . . . . : *NONE
Print separator page . . . . . . . . . : *NO
User defined option . . . . . . . . . . : *NONE
User defined object:
Object . . . . . . . . . . . . . . . . : *NONE
Library . . . . . . . . . . . . . . . :
Object type . . . . . . . . . . . . . . :
User driver program . . . . . . . . . . : *NONE
Library . . . . . . . . . . . . . . . :
Spooled file ASP . . . . . . . . . . . : *SYSTEM
Text description . . . . . . . . . . . : *BLANK
* * * * * E N D O F L I S T I N G * * * * *
```