

VERSION 18.0.0  
APRIL 2023  
702P09002

# Xerox® FreeFlow® VI Compose Software

User Guide

© 2023 Xerox Corporation. All rights reserved. XEROX®, FreeFlow®, FreeFlow Makeready®, VIPP®, and GlossMark® are trademarks of Xerox Corporation in the United States and/or other countries. Other company trademarks are acknowledged as follows:

Adobe PDFL - Adobe PDF Library Copyright © 1987-2021 Adobe Systems Incorporated.

Adobe PDF Converter – Adobe PDF Converter Library Copyright © 2021 Adobe Systems Incorporated.

Adobe®, the Adobe logo, Acrobat®, the Acrobat logo, Acrobat Reader®, Distiller®, Adobe PDF JobReady™, InDesign®, PostScript®, and the PostScript logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States and/or other countries. All instances of the name PostScript in the text are references to the PostScript language as defined by Adobe Systems Incorporated unless otherwise stated. The name PostScript is used as a product trademark for Adobe Systems implementation of the PostScript language interpreter, and other Adobe products. Copyright 1987-2021 Adobe Systems Incorporated and its licensors. All rights reserved. Includes Adobe® PDF Libraries and Adobe Normalizer technology.

Intel®, Pentium®, Centrino®, and Xeon® are registered trademarks of Intel Corporation. Intel Core™ Duo is a trademark of Intel Corporation.

Intelligent Mail® is a registered trademark of the United States Postal Service.

Macintosh®, Mac®, OS X®, and macOS® are registered trademarks of Apple, Inc., registered in the United States and other countries. Elements of Apple Technical User Documentation used by permission from Apple, Inc.

Novell® and NetWare® are registered trademarks of Novell, Inc. in the United States and other countries. Oracle® is a registered trademark of Oracle Corporation Redwood City, California.

PANTONE™ and other Pantone Inc. trademarks are the property of Pantone Inc. All rights reserved. QR Code™ is a trademark of Denso Wave Incorporated in Japan and/or other countries.

TIFF® is a registered trademark of Aldus Corporation.

The Graphics Interchange Format© is the Copyright property of CompuServe Incorporated. GIFSM is a Service Mark of CompuServe Incorporated.

Windows®, Windows® 10, Windows® 11, Windows Server® 2016, Windows Server® 2019, Windows Server® 2022, and Internet Explorer are trademarks of Microsoft Corporation; Microsoft® and MS-DOS® are registered trademarks of Microsoft Corporation.

All other product names and services mentioned in this publication are trademarks or registered trademarks of their respective companies. They are used throughout this publication for the benefit of those companies, and are not intended to convey endorsement or other affiliation with the publication.

Companies, names, and data used in examples are fictitious unless otherwise noted.

While every care has been taken in the preparation of this material, no liability is accepted by Xerox Corporation arising out of any inaccuracies or omissions.

Changes are made periodically to this document. Changes, technical inaccuracies, and typographical errors are corrected in subsequent editions.

Produced in the United States of America.

BR38584

# Contents

FreeFlow VI Compose Licensing and Installation.....	9
Getting Started.....	10
Program Download.....	10
Specialty Imaging and Barcode Font Download.....	10
VI Suite Customer Forum.....	11
Demonstration and Production Modes.....	12
Demonstration Mode.....	12
Production Mode.....	12
Installation Overview.....	13
Install the VI Compose Software.....	13
Obtain a VI Compose License.....	13
Complete the Installation.....	14
Font Installation.....	15
VI Compose Installation on FreeFlow Print Server (Solaris) DFEs.....	16
Customized File Backup.....	16
Software Installation.....	16
License Installation.....	19
VI Compose Upgrade Instructions for Windows FreeFlow Print Server Devices.....	20
Customized File Backup.....	20
Software Upgrade Installation.....	21
VI Compose Installation on the EFI DFEs Driving Xerox Print Engines.....	24
Software Installation.....	24
License Installation.....	26
Installing the Makeready Applications Program Interface (API).....	29
Makeready Software Installation.....	29
Installing the Makeready Applications Program Interface (API).....	29
Configuring the Makeready Applications Program Interface (API).....	30
VI Compose Installation on DocuPrint NPS Devices.....	42
Customized File Backup.....	42
Software Installation.....	43
License Installation.....	44
Variable Information Suite Documentation.....	47
Intended Audience.....	48
Documentation Overview.....	49
VIPP® and VI Compose Overview.....	51
Basic Concepts.....	52
VIPP® and PostScript.....	52
VIPP®-enabled Devices.....	52
Repositories.....	53
VI Projects.....	53
SAP Device Type for VI Compose.....	57

Using PDF Resources with VIPP® APPE, PDF/VT, and Embed EPS Options.....	58
Printing the VIPP® Application on a Legacy PostScript RIP .....	58
Printing the VIPP® Application to a PostScript Printer.....	60
Printing the VIPP® Application on an APPE RIP.....	60
Printing the VIPP® Application to FFPS APPE.....	61
Producing a PDF from a VIPP® Application, VI eCompose, or PDF Export .....	63
Adobe Acrobat and Adobe Reader Information.....	63
Printing PDF Files with Referenced XObjects.....	64
VIPP® and Variable Information Suite Applications.....	65
VI Compose Functionality .....	65
FreeFlow® VI Design Pro .....	67
FreeFlow® VI eCompose .....	68
FreeFlow® VI Design Express.....	70
FreeFlow® VI Explorer.....	70
FreeFlow® VIPP® Manage .....	71
FreeFlow® VI Compose Open Edition .....	72
VIPP® Data Streams .....	73
Native Mode .....	75
Line Mode.....	76
Listing .....	76
Record Processing Entry .....	76
Database Mode .....	78
Selecting Database Field Names .....	79
XML Mode .....	80
XML Data Structure Overview.....	80
XML Variables .....	80
Printing XML Variables .....	82
Stripping Blank and Control Characters.....	82
Prefixed Delimited Transactional Data.....	83
Linking the Data with the JDT, DBM, or XJT.....	84
Record Structures .....	85
Record Length.....	85
Vertical Control .....	85
Line Printer and Field Delimited Data .....	86
Prefix.....	86
Record Structures and Mode/Presentation Relationship .....	87
Functions.....	88
Conditional Processing .....	90
Dynamic Boxes .....	91
Data Driven Graphics .....	92
Native Mode Prefix.....	93
Distribution List.....	94
End of File .....	95
Coding Examples.....	96

Native Mode Coding .....	96
Line Mode File .....	96
Job Descriptor Ticket .....	96
Database File.....	97
Data Base Master file .....	98
XML Mode Coding .....	98
XML .....	99
VIPP® Resources .....	101
Resource Types .....	102
VI Project .....	102
VI Compose and Setup Files .....	103
Fonts.....	104
Forms .....	110
Segments.....	110
Images.....	110
PDF Files as Resources .....	110
Job Descriptor Tickets.....	111
XML Job Tickets.....	111
Data Base Masters.....	111
Distribution Lists.....	111
Text Files.....	111
FreeFlow Makeready RDO Files.....	112
Resource Access and Management.....	113
Managing VIPP® Resources .....	113
VI Compose Files and Utilities .....	115
Link and Setup Files .....	116
xgfunix.run link file (UNIX and Windows FFPS).....	116
xgfdos.run link file (VI Windows products) .....	121
General Defaults Setup File.....	122
Utilities and Demonstration Files.....	125
xgfprint .....	125
VI Project Container Filter .....	126
VIPP® Manage.....	126
VI Compose version information .....	126
License Request forms .....	126
VIPP® demonstration files .....	126
Demonstration file list .....	127
VIPP® File Examples .....	133
Example 1: Bank Statement with Images Using Native Mode.....	134
Print File Example .....	134
Form File Example .....	135
Example 2: Sales Report Using Line Mode.....	137
Print File Example .....	137

Job Descriptor Ticket File Example .....	138
Example 3: Sales Report Using RPE Line Mode .....	140
Print File Example .....	140
Form File Example .....	141
Job Descriptor Ticket File Example .....	142
Example 4: Invoice Using RPE Prefixed Line Mode .....	144
Print File Example .....	144
Form File Example .....	145
Job Descriptor Ticket File Example .....	148
Example 5: Mailing Using Database Mode .....	151
Print File Example .....	151
Data Base Master File Example .....	151
Example 6: Bank Statement Using Dynamic Boxes and Data-Driven Graphics .....	154
Print File Example .....	154
Job Descriptor Ticket File Example .....	156
Standard Lists, Tables, Keys, and Attributes .....	163
Standard Font Lists and Encoding Tables .....	164
Re-encoded Font List File .....	164
Macintosh Character Set .....	164
PostScript-Encoded Font List File .....	164
PC-8 Encoding Table File .....	165
pcsun Encoding Table File .....	165
ISO Latin 1 Encoding Table .....	165
UTF-8 Encoding Table .....	165
win1252 Encoding Table .....	166
Standard GEPkeys, PATkeys, BATkeys, and Colorkeys .....	167
Graphic Element Property Keys and Colorkeys .....	167
Pattern Keys .....	167
Standard Background Attributes .....	167
Solid Coated and Uncoated Custom Colors .....	167
Predefined PCC Tables .....	168
Predefined Multi-Up Definitions .....	169
Predefined Multiple-Byte Encoding Table .....	170
Printing with VI Compose .....	171
Decomposition Services .....	172
Referencing Decomposed Forms Using a Generic Method .....	172
Referencing forms Decomposed in DocuPrint NPS .....	172
Referencing Forms Decomposed in the FreeFlow Print Server .....	173
Referencing Decomposed Images in DocuPrint NPS Devices .....	173
Referencing Decomposed Images in the FreeFlow Print Server .....	174
Decomposition Services Hints and Tips .....	174
Using VI Compose on Specific Printers .....	176
VIC Considerations for iGen .....	176
VIC considerations for DocuPrint NPS .....	176

VIC Considerations for FreeFlow Print Server .....	179
VIC Considerations for DocuColor Devices .....	183
VIC considerations for the Office Environment .....	183
Storing VIPP® Resources on Office Printers .....	184
VIPP® Command Differences .....	185
VI Compose and FreeFlow Makeready .....	187
VI Compose/Makeready API overview .....	187
How VI Compose calls the RDO file .....	188
VIPP® Text .....	189
Using an RDO File in a VIPP® Job .....	192
LCDS Migration .....	194
General Description .....	194
Basic LCDS .....	195
LCDS and VIPP® resources .....	195
LCDS and VIPP® Commands .....	196
Conversion Examples .....	197
LCDS Migration Hints and Tips .....	200
Specialty Imaging with VIPP® .....	203
Specialty Imaging Features .....	208
Programming Support .....	210
Using Specialty Imaging .....	211
Download and Install the Specialty Imaging Fonts .....	211
Set the FreeFlow Print Server to Use Specialty Imaging .....	211
FFPS Printer Support Restrictions .....	214
Demonstration Jobs .....	216
Using Specialty Imaging Effects .....	217
Limitations .....	221
Design Considerations for Correlation Mark .....	223
Correlation Mark Text Limitations and Requirements .....	224
Correlation Mark Color Swatches .....	224
Design Considerations for FluorescentMark Color .....	227
FluorescentMark Limitations and Requirements .....	227
Designing Documents with FluorescentMark Colors .....	232
Two-Layer Fluorescent Text .....	235
Design Considerations for GlossMark® Text .....	236
GlossMark Text Limitations and Requirements .....	236
GlossMark Colorkey Table .....	237
Designing Documents with GlossMark Text .....	237
Design Considerations for Infrared Color .....	239
Infrared Colorkeys .....	239
Designing Documents with Infrared Color .....	239
Infrared Colorkey Tables .....	239
Designing Documents with Infrared Color .....	241
Design Considerations for MicroText Font .....	244
MicroText Font Limitations and Requirements .....	244

MicroText Font Table .....	245
Designing Documents with MicroText Fonts.....	246
Design Considerations for Void Pantograph.....	247
VI Compose Services .....	249
Normalization.....	250
Demographics .....	252
File samples .....	255
Travel_us.vpr.....	255
ACCLOG report 1 .....	257
ACCLOG report 2.....	258
Demographics report 1 .....	259
Demographics report 2.....	259

# FreeFlow VI Compose Licensing and Installation

This chapter contains:

Getting Started.....	10
VI Suite Customer Forum .....	11
Demonstration and Production Modes.....	12
Installation Overview .....	13
Font Installation .....	15
VI Compose Installation on FreeFlow Print Server (Solaris) DFEs .....	16
VI Compose Upgrade Instructions for Windows FreeFlow Print Server Devices.....	20
VI Compose Installation on the EFI DFEs Driving Xerox Print Engines .....	24
Installing the Makeready Applications Program Interface (API) .....	29
VI Compose Installation on DocuPrint NPS Devices.....	42

*FreeFlow® VI Compose Licensing and Installation* provides information to begin installation of the VI Compose license and software. These topics are discussed:

- Getting started
- VI Suite Customer Forum
- Demonstration and Production modes
- Installation overview
- Font installation
- VI Compose Installation on FreeFlow Print Server (Solaris) DFEs
- VI Compose Upgrade Instructions for Windows FreeFlow Print Server Devices
- VI Compose Installation on the EFI DFEs (Driving Xerox print engines)
- VI Compose / Makeready API Installation
- VI Compose Installation on DocuPrint NPS Devices

## Getting Started

Before you begin, first download the software from the Xerox website. Download instructions are provided here. After you have downloaded the appropriate files you can proceed to the installation instructions that follow.

### PROGRAM DOWNLOAD

FreeFlow VI Compose can be delivered preloaded on the device, or it can be downloaded as follows:

Verify that VI Compose is loaded on the device by locating the xgf directory on the device hard drive. If the xgf directory is not present, VI Compose is not loaded onto the device. Download the program, then refer to [Installation overview](#) to locate installation information for the device.

To download the VI Compose software electronically, go to [www.xerox.com/support](http://www.xerox.com/support), then select **Software & Platforms**. Select **FreeFlow > FreeFlow Variable Information Suite**, then select **Software and Solutions**. If necessary, to display the correct installer file, select the operating system type appropriate for your target platform. Under Utilities & Applications, download the FreeFlow VI Compose 18.0.0 installer file.



**Note: Program delivery** Programs are delivered in .iso or .exe format. These files can be downloaded directly to the target device. When the device does not have Internet access, copy the .exe file on the target device, or use the .iso file to burn a CD from which the program can be installed.

### SPECIALTY IMAGING AND BARCODE FONT DOWNLOAD

To download specialty imaging fonts, go to [www.xerox.com/support](http://www.xerox.com/support), select **Software & Platforms > FreeFlow > FreeFlow Variable Information Suite > Software and Solutions**, then click on one of the fonts to download.



**Note:** When downloading fonts, you will be directed to review an End User License Agreement. In order to download the fonts you must review and accept the End User License Agreement. If you do not accept the End User License Agreement, you will exit from the font download page.

## VI Suite Customer Forum

Xerox hosts a Community Support Forum. The VI Suite Customer forum is now part of this larger support forum, allowing you to post and review information about Xerox products and services all from one location. Refer to this customer forum community: <http://VIPSupport.xerox.com>.

## Demonstration and Production Modes

You can install the VI Compose software in two modes, Demonstration and Production.

### DEMONSTRATION MODE

When VI Compose is installed without a production license file, it is considered to be installed in demonstration mode. Demonstration mode is a full-featured version of VI Compose with two exceptions:

- Customer support is not available
- Page volume limitations are imposed

VI Compose limits the number of pages that can be printed when running in Demonstration mode. The limit is device-dependant and varies between 10 and 200 pages. VIPP® jobs that exceed this page volume limit will abort with the following error message:

```
ERROR: undefined  
OFFENDING COMMAND: selected pages 0 n
```

Where:

**n** is the page volume limit for that device.

This error message indicates that the demonstration page capacity for the device has been exceeded. On some DocuColor printers, the error does not appear. Instead, jobs stop when the demonstration limit is reached. The limit is 10, 57, 150 or 200 pages and depends upon the device.

### PRODUCTION MODE

To enable the Production mode, install a valid VI Compose license on the device. In production mode VIPP® jobs can be run without any page limitations. Contact a Xerox representative for more information about running VI Compose in Production mode.

## Installation Overview

The following sections provide the general steps required to:

- [Install the VI Compose software](#)
- [Obtain a VI Compose License](#)
- [Complete the installation](#)

### INSTALL THE VI COMPOSE SOFTWARE

Use this table to locate the software installation instructions for the device.



Note: You cannot install VI Compose on a Virtual machine, no support is provided for the VM environment.

WHEN INSTALLING VI COMPOSE ON THIS DEVICE:	GO TO THIS CHAPTER:
DocuPrint NPS (monochrome and color)	<a href="#">VI Compose Installation on DocuPrint NPS Devices</a>
All FreeFlow Print Server-based controllers including FreeFlow Print Server controllers for monochrome and color devices (including iGen)	<a href="#">VI Compose Installation on FreeFlow Print Server (Solaris) DFEs</a>
External EFI controllers	<a href="#">VI Compose Installation on the EFI DFEs (Driving Xerox print engines)</a>
FreeFlow Makeready	<a href="#">VI Compose / Makeready API Installation</a>

### OBTAIN A VI COMPOSE LICENSE

An existing license from an earlier version of the software will not enable this new version. A new license is required for every major software release whether it is an upgrade or a new install.

Use the following instructions to request a new license.

#### Windows Licensing

For either an UPGRADE to an existing installation or for a NEW installation:

1. Locate the Get Host ID utility, which can be found at **Start Menu > All Programs > FreeFlow VI Compose program group**.
2. Click on the **Get Host ID** utility, this will open a Window displaying the HostID and Version Number. You will need this information when talking to a Xerox representative.
3. Contact the Xerox License Administrator.

#### UNIX / Solaris - New Installation

1. Install the VI Compose software.

2. Locate the ..\xgf directory on the hard drive where VI Compose was installed.
  - When A4 paper size was selected during the software installation, print the vipplrfxe.ps file.
  - When USLetter paper size was selected during the software installation, print the vipplrfus.ps file
3. Contact a Xerox representative.

### **UNIX / Solaris - Upgrade to an Existing Installation**

1. Locate the ..\xgf directory on the hard drive where VI Compose was installed.
  - When A4 paper size was selected during the software installation, print the vipplrfxe.ps file.
  - When USLetter paper size was selected during the software installation, print the vipplrfus.ps file
2. Contact a Xerox representative.



Note: When updating the FFPS, you can use the FFPS Update Manager to update the VI Compose software.

### **COMPLETE THE INSTALLATION**

Follow the instructions found in the device-specific chapters listed in [Install the VI Compose software](#).

## Font Installation

During installation of VI Compose on VIPP® enabled devices, when you install VDP, VIE, and VIEC, font installation options are provided. Install only the fonts you use. When you avoid installing unnecessary fonts, you save time and use less of the available memory on your VIPP®-enabled device. Specific information about specialty imaging fonts is located in [Specialty Imaging with VIPP](#).

Support of OTF/TTF fonts is PostScript-interpreter dependent. Some PostScript interpreters support OTF/TTF fonts. If the PostScript interpreter where VI Compose is installed supports OTF and TTF fonts, you can use the fonts in VIPP® code.

You can place an OTF or TTF font in the same directory where PostScript fonts usually reside. When you copy the OTF or TTF font files to the font directory, delete the **.otf** and **.ttf** file name extensions. After you copy the fonts to the font directory, the fonts can be referenced from within the VIPP® code, using the font file name, not the internal PostScript font name (/FontName).

As an example, to use the font RosewoodStd-Regular.otf:

1. Copy C:/WINDOWS/Fonts/RosewoodStd-Regular.otf to:
  - C:\Program Files (x86)\Xerox\VIPP\vide\fonts\RosewoodStd-Regular for VDP
  - C:\Program Files (x86)\Xerox\VIPP\vviewer\fonts\RosewoodStd-Regular for VIE
  - /opt/XRXnps/resources/ps/fonts/RosewoodStd-Regular for FFPS (UNIX systems)
  - x:\efi\server\adobe\fonts\RosewoodStd-Regular for EFI
  - x:\CXP6000\General\RIP\fonts\RosewoodStd-Regular
  - C:\Program Files (x86)\Xerox\VIPP\xtvp\bin\normalizer\fonts for VIEC
2. Use the font as a PostScript font in the VIPP® code. Use the font file name as in one of these options:
  - Option 1, if no re-encoding is needed:
 

```
/RosewoodStd-Regular 20 SETFONT
```
  - Option 2, if re-encoding is needed:
 

Use SETENCODING:

```
[ /NRWSR /RosewoodStd-Regular ] (sun8) SETENCODING
```

Or, add the following line to encoding/fontlist:

```
/NRWSR /RosewoodStd-Regular
```

Select the font with the VIPP® font name:

```
/NRWSR 20 SETFONT
```

## VI Compose Installation on FreeFlow Print Server (Solaris) DFEs

Use these instructions to install the VI Compose software and license on all devices driven by the FreeFlow Print Server.

### CUSTOMIZED FILE BACKUP

During an upgrade customized VIPP® files located in `/usr/xgf/src` and `/usr/xgf/encoding`, are overwritten with new files of the same name. The system automatically copies and renames those directories as a form of backup for customized files. The backup versions of the original directories are `/usr/xgf/src.$OLDVER` and `/usr/xgf/encoding.$OLDVER`, where `$OLDVER` represents the previous version of VI Compose. When customized files exist in either of the original directories, it is your responsibility to merge the customized contents of those files with the new versions.

Xerox recommends that additional copies of all customized files be stored external to the system or printed as hardcopy backup. This is extremely important if the customized file is not included in the list below.

<code>xgf/src/arb.def</code>	Arabic configuration file
<code>xgf/src/cjk.def</code>	Asian configuration file
<code>xgf/src/xgf</code>	VI Compose startup file
<code>xgf/src/xgf.bat</code>	Background attribute (bat) keys
<code>xgf/src/xgf.def</code>	VI Compose system defaults
<code>xgf/src/xgf.gep</code>	Color and graphical element property (gep) keys
<code>xgf/src/xgf.mup</code>	Multi-Up definitions
<code>xgf/src/xgf.pcc</code>	Standard pcc definitions
<code>xgf/src/xgfunix.run</code>	VI Compose startup definition file
<code>xgf/encoding/fontlist</code>	Generic format encoding font list
<code>xgf/encoding/nullfl</code>	Null encoding font list



Note: Any previous customization of the VI Compose environment will not be valid in a new upgrade. Copying the old files has the potential of rendering the new functionality unusable. Any customization will have to be repeated on the new files after a successful upgrade.

### SOFTWARE INSTALLATION

This section contains software installation and upgrade instructions for VI Compose on FreeFlow Print Server devices.

To install VI Compose software to FFPS systems supporting the FFPS Update Manager, use the Update Manager to install the latest available VI Compose software. Refer to your FFPS documentation for more information about the FFPS Update Manager. If not using the FFPS Update Manager then follow the process below.

## Install VI Compose on a FreeFlow Print Server Device

To install VI Compose software on a FreeFlow Print Server device:

1. From [www.xerox.com/support](http://www.xerox.com/support), download the FreeFlow VI Compose 17 software .iso file. For more information, refer to [Program download](#).
2. Burn the downloaded .iso file to a CD.
3. Open a terminal session on the workstation where VI Compose software is installed.
4. Log in as root.
5. Insert the CD that contains the .iso file into the Solaris drive.
6. To access the software directory, type `cd /cdrom/vippXX/Solaris/[X86|Sparc]`, where XX is the software version.



Note: If you are using a .tar file, you can download the DSPVIPxx.i386.tar file or the sparc.tar file. Move the tar file to the /var/tmp directory. At the root prompt, enter the following text: `tar -xvf DSPVIPxx.i386 (or sparc) .tar`.

7. Execute the installation script command. Enter:

```
sh ./vippinstall
```

The installation script checks the environment setup and looks for an existing VI Compose software package. When the system finds a software package, the following messages appear:

```
VIPxx package was found. Installation cannot continue unless this existing
package is removed first.
```

```
May I uninstall this existing package now?
```

8. To uninstall the existing software package, enter **y**.

Directories with customized files are backed up during this step. The installation script provides the specifics of the backup.

```
Removal of <VIPxx> was successful.
```

```
Welcome to the VI Compose package installer for FreeFlow Print Server [sparc|
i386].
```

```
This process installs the package for VI Compose xx s/w release.
```

```
Build VIPP xx: build date and time
```

```
NOTICE:
```

```
This stage of the installation can be exited at any time by typing CTRL-C.
```

9. To continue, press **Enter**.

10. Installation requires input for the Default Media setting.

- For A4 media, enter 1.
- For the default USLetter media, enter 2.

This message appears:

```
Default Media Setting: USLetter
```

```
Is this setting correct?
```

11. When the setting is correct, enter **y**. When the setting is incorrect, enter **n**.

12. When this message appears, to continue, press **Enter**.

WARNING: This stage of the installation must NOT be interrupted. Otherwise, installation of VI Compose s/w is corrupted.

These system messages appear:

```
Processing package instance <VIPxx> from /cdrom/vipxx/Solaris/[X86|Sparc]
```

```
Installing Xerox FreeFlow Print Server VIPP Update xx as <VIPxx>
```

```
Installation of <VIPxx> was successful.
```

```
Exiting vippinstall for package VIPxx with status 0.
```

13. Restart the FreeFlow Print Server.

14. Validate the installation.

### VIPP® Golden Job

To validate the installation or upgrade run the VIPP® Golden Job.

1. At the FreeFlow Print Server Print Services window, select **Logon > Logon** and log on as System Administrator.
2. Program the paper trays with the paper stocks shown in the table in the next step.

Select the appropriate paper size in the Size area. When the A4 version of VI Compose was selected during the software installation, select **A4**. When the USLetter version was selected during installation, select **USLetter**.

Ensure that White is selected in the Color area.

In the Type area, select **Custom**, then enter the color name from the Type column of the following Paper Tray Setup table.

3. Select **OK**.

Repeat the steps described in the previous step for each paper tray listed in this table:

TRAY	SIZE	COLOR	TYPE	STOCK NAME
1	US Letter or A4	White	Yellow	Unspecified
2	US Letter or A4	White	Green	Unspecified
3	US Letter or A4	White	Plain	Unspecified

4. Load paper trays 1, 2, and 3 with the paper stocks programmed in the previous step.

- From a terminal window, enter:

```
cd /usr/xgf/demo
```

- To print the Golden Job, enter this command:

```
print -d <queue name> goljobv
```

Or, use the SUN CDE file manager to print the data file.

The completed job should consist of 43 single-sided pages. Pages 1 and 43 are printed on Tray 1 stock (yellow), pages 8, 10, and pages 25 through 27 are printed on Tray 2 stock (green). The remaining pages are printed on Tray 3 stock (plain).

## LICENSE INSTALLATION

To run in production mode, ensure that VI Compose is fully licensed. Use these steps to determine whether the license has been installed, and to install it when it has not been installed.

- Determine whether a VI Compose license has been installed; at the UNIX workstation, enter: **pkginfo | grep vip**

Look for package name **XRXvipp**. When the package is found a license has been installed and no further action is necessary. When not, follow these steps to acquire a license:

- Obtain the Host ID number - At a UNIX command tool window enter **hostid** to display the number
- Call a Xerox representative to obtain the license file.

The license will be issued in an email. Upon receipt of the license string, follow the steps outlined below.

- Log on as the FreeFlow Print Server Administrator. In the System Administrator window select **Setup > <Feature Licenses>**

- In the License Manager Window select the **Variable Information Production Printware** option; then:

- Select the **License** pull-down menu option
- Select **Load License File...**
- Follow the GUI options to locate the license file
- Select **<OK>**

- Verify that the license string has been correctly installed by entering:

```
cd /usr/xgf/demo
```

```
print -p <queue name> imgdemo.nm
```

- When the PostScript Error Page option is enabled and the license string was not installed correctly or has expired, the message "VIPP\_license\_failed" will appear on a PostScript error page. Restart the system. From the FreeFlow Print Server Print Services window, select **System > Restart**.

## VI Compose Upgrade Instructions for Windows FreeFlow Print Server Devices

Use these instructions to upgrade the VI Compose software on all devices driven by the Windows FreeFlow Print Server (FFPS).

### CUSTOMIZED FILE BACKUP

During an upgrade, customized VIPP® files located in %XPS\_HOME2%\resources\xgf\src and %XPS\_HOME2%\resources\xgf\encoding are overwritten with new files of the same name. To prevent the loss of data in those customized files, the installer automatically copies them to %XPS\_HOME2%\resources\xgf\_backup before the new files are installed. When customized files exist in the xgf\_backup directory, it is your responsibility to merge the customized contents of those files with the new versions.



**Note:** If the resources\xgf directory cannot be found at C:\Program Files (x86)\Xerox-PS\, the current default setting for the environment variable XPS\_HOME2, obtain the value of XPS\_HOME2 as follows: Click on the Windows Start icon, right click on Computer, select Properties, click on Advanced system settings, select the Advanced tab, click on the Environment Variables... button and scroll down in the System variables menu to locate XPS\_HOME2.

Typically, customization is done in one of the following files. Even though the following files will be backed up during an upgrade installation, Xerox recommends that additional copies of all customized files be stored external to the system or printed as hardcopy backup. This is extremely important if the customized file is not included in the list below.

xgf\src\arb.def	Arabic configuration file
xgf\src\cjkl.def	Asian configuration file
xgf\src\xgf	VI Compose startup file
xgf\src\xgf.bat	Background attribute (BAT) keys
xgf\src\xgf.def	VI Compose system defaults
xgf\src\xgf.gep	Color and graphical element property (GEP) keys
xgf\src\xgf.mup	Multi-Up definitions
xgf\src\xgf.pcc	Standard pcc definitions
xgf\src\xgfunix.run	VI Compose startup definition file
xgf\encoding\fontlist	Generic format encoding font list
xgf\encoding\nullfl	Null encoding font list



**Note:** Any previous customization of the VI Compose environment will not be valid in a new upgrade. Copying old files has the potential of rendering new functionality unusable. Any customization will have to be repeated in the new files after a successful upgrade.

## SOFTWARE UPGRADE INSTALLATION

This section contains software upgrade installation instructions for VI Compose on Windows FFPS devices.

On Windows FFPS supporting the FFPS Update Manager, use the Update Manager to upgrade to the latest available VI Compose software. Refer to your Windows FFPS documentation for additional information. If not using the Update Manager then follow the process below:

1. Download the VI Compose .iso file to Windows FFPS and either burn the .iso to a CD or extract its contents using a utility such as WinRAR.
2. Go to the Windows\_FFPS folder. You see three files: updateVIPP.bat, XRXxgf.msi, and readme.txt which is the most up-to-date copy of these instructions in text format.
3. Stop the Windows FFPS software if it is currently running.

- Execute the updateVIPP.bat installation script via a Windows PowerShell or Command Prompt with administrative privileges:

```
C:\>. \updateVIPP.bat
```

The installation script will check the environment setup and look for an existing version of VI Compose software. If the VI Compose software wasn't previously installed via the Windows FFPS installer, the following message will be displayed and the upgrade installation process will terminate:

```
"Update Failed."
"FFPS must have been installed on this system prior to running updateVIPP.bat"
```

When VI Compose software is found, the upgrade installation will proceed and messages will be displayed indicating that the above mentioned files in `xgf\src` and `xgf\encoding` are being backed up.

```
PS C:\ffps> .\updateVIPP.bat
C:\ffps>ECHO "Backing up files..."
"Backing up files..."
C:\ffps>IF "C:\Program Files (x86)\Xerox-PS\" == "" GOTO NO_PREVIOUS_INSTALL
C:\ffps>REM Create backup directories
C:\ffps>MKDIR "C:\Program Files (x86)\Xerox-PS\resources\xgf_backup\encoding"
A subdirectory or file C:\Program Files (x86)\Xerox-PS\resources\xgf_backup\encoding already exists.
C:\ffps>MKDIR "C:\Program Files (x86)\Xerox-PS\resources\xgf_backup\src"
A subdirectory or file C:\Program Files (x86)\Xerox-PS\resources\xgf_backup\src already exists.
C:\ffps>REM Copy UER file to backup
C:\ffps>COPY "C:\Program Files (x86)\Xerox-PS\resources\xgf\UER" "C:\Program Files (x86)\Xerox-PS\resources\xgf_backup"
1 file(s) copied.
C:\ffps>REM Copy encoding files to backup
C:\ffps>COPY "C:\Program Files (x86)\Xerox-PS\resources\xgf\encoding\fontlist" "C:\Program Files (x86)\Xerox-PS\resources\xgf_backup\encoding"
1 file(s) copied.
C:\ffps>COPY "C:\Program Files (x86)\Xerox-PS\resources\xgf\encoding\nullfl1" "C:\Program Files (x86)\Xerox-PS\resources\xgf_backup\encoding"
1 file(s) copied.
C:\ffps>REM Copy src files to backup
C:\ffps>COPY "C:\Program Files (x86)\Xerox-PS\resources\xgf\src\xgf" "C:\Program Files (x86)\Xerox-PS\resources\xgf_backup\src"
1 file(s) copied.
C:\ffps>COPY "C:\Program Files (x86)\Xerox-PS\resources\xgf\src\xgfunix.run" "C:\Program Files (x86)\Xerox-PS\resources\xgf_backup\src"
1 file(s) copied.
C:\ffps>COPY "C:\Program Files (x86)\Xerox-PS\resources\xgf\src\xgf.bat" "C:\Program Files (x86)\Xerox-PS\resources\xgf_backup\src"
1 file(s) copied.
C:\ffps>COPY "C:\Program Files (x86)\Xerox-PS\resources\xgf\src\xgf.gcp" "C:\Program Files (x86)\Xerox-PS\resources\xgf_backup\src"
1 file(s) copied.
C:\ffps>COPY "C:\Program Files (x86)\Xerox-PS\resources\xgf\src\xgf.def" "C:\Program Files (x86)\Xerox-PS\resources\xgf_backup\src"
1 file(s) copied.
C:\ffps>COPY "C:\Program Files (x86)\Xerox-PS\resources\xgf\src\arb.def" "C:\Program Files (x86)\Xerox-PS\resources\xgf_backup\src"
1 file(s) copied.
C:\ffps>COPY "C:\Program Files (x86)\Xerox-PS\resources\xgf\src\cjk.def" "C:\Program Files (x86)\Xerox-PS\resources\xgf_backup\src"
1 file(s) copied.
C:\ffps>COPY "C:\Program Files (x86)\Xerox-PS\resources\xgf\src\xgf.pcc" "C:\Program Files (x86)\Xerox-PS\resources\xgf_backup\src"
1 file(s) copied.
C:\ffps>COPY "C:\Program Files (x86)\Xerox-PS\resources\xgf\src\xgf.mup" "C:\Program Files (x86)\Xerox-PS\resources\xgf_backup\src"
1 file(s) copied.
```

When the back-up is complete, updateVIPP.bat will invoke `XRXxgf.msi` to replace/overwrite all of the files in the `xgf` directory and subdirectories. Messages will be displayed during this process.

```

C:\ffps>ECHO "Beginning update..."
"Beginning update..."
C:\ffps>REM Run the UIPP installer; REINSTALLMODE=vamus forces all files to be reinstalled
C:\ffps>msiexec.exe -i XRXxgf.msi REINSTALL=ALL REINSTALLMODE=vamus
C:\ffps>REM Run ModifyXgfunixFiles.exe - Modify the paths in xgfunix.run
C:\ffps>START "" /D "C:\Program Files (x86)\Xerox-PS\resources\xgf\src" /B /WAIT "ModifyXgfunixFiles.exe"
XPS_UIPP_HOME:
  C:\Program Files (x86)\Xerox-PS/
XPS_UIPP_DATA:
  C:\ProgramData\Xerox-PS/
UIPP_XGF_DIRPATH:
  C:\Program Files (x86)\Xerox-PS/resources/xgf/
UIPP_XGF_STARTUPFILE:
  C:\ProgramData\Xerox-PS\data\ripService\ps\Startup\startupUIPP.ps
UIPP_XGF_SPOOLFILE:
  C:\ProgramData\Xerox-PS\data\xgfc\temp\xgfspool
xgf file path:
  C:\Program Files (x86)\Xerox-PS/resources/xgf/src/xgf
xgfunix.run file path:
  C:\Program Files (x86)\Xerox-PS/resources/xgf/src/xgfunix.run

File C:\Program Files (x86)\Xerox-PS/resources/xgf/src/xgfunix.run exists

File C:\Program Files (x86)\Xerox-PS/resources/xgf/src/xgf exists

File C:\ProgramData\Xerox-PS\data\ripService\ps\Startup\startupUIPP.ps exists
C:\ffps>GOTO DONE
C:\ffps>ECHO "Update completed."
"Update completed."
PS C:\ffps>

```



Note: The Locale registry key is used for determining whether to install the US Letter or A4 version of the software.

- Restart the Windows FFPS software and follow the steps in the next section to validate the VI Compose software upgrade installation.

### VIPP® Golden Job

To validate the installation, run the VIPP® Golden Job. Note that some knowledge of Windows FFPS is required.

- At the Xerox FFPS window, log on as System Administrator.
- Program paper trays 1, 2, and 3 with the paper stocks shown in the table below.

TRAY	SIZE	COLOR	TYPE	STOCK NAME
1	US Letter or A4	White	Yellow	Unspecified
2	US Letter or A4	White	Green	Unspecified
3	US Letter or A4	White	Plain	Unspecified

- Load paper trays 1, 2, and 3 with the paper stocks programmed in the previous steps.
- Submit the demo job `.. \xgf\demo\go1jobv`

Ensure that the completed job is consist of 43 single-sided pages. Pages 1 and 43 are printed on Tray 1 stock (yellow); pages 8, 10, and 25 through 27 are printed on Tray 2 stock (green), and the remaining pages are printed on Tray 3 stock (plain).

## VI Compose Installation on the EFI DFEs Driving Xerox Print Engines

Use these instructions to install the VI Compose software and license on a EFI DFE.



Note: When you upgrade or uninstall this software, ensure that you take back up of any customized files that can be required at a later date.

### SOFTWARE INSTALLATION

This section contains instructions for installing VI Compose software on the EFI DFEs. Refer to [Possible warning and error dialogs](#) for more information.

To install VI Compose software to FFPS systems supporting the FFPS Update Manager, use the Update Manager to install the latest available VI Compose software. Refer to your FFPS documentation for additional information about the FFPS Update Manager. If not using the FFPS Update Manager then follow the process below.

The VI Compose software can be downloaded, refer to [Program download](#) for more information. Once downloaded, the VI Compose .iso file must be burnt to a CD to install to the EFI system.

To install VI Compose software on the EFI DFE for DocuColor systems:

1. Insert the VI Compose CD. If the VI Compose Setup Program does not launch automatically, use one of these methods:
  - The Run window:  
Select **Start > Run** to access the Run window, then click **Browse** to make sure that the CD drive, or alternate network location, and `XeroxFreeFlowVCompose.exe` are listed in the Open field (for example, `E:\windows\XeroxFreeFlowVCompose.exe`). When the correct drive and file name are selected, click **OK**.
  - Windows Explorer:  
Access root on the VI Compose CD or the alternate location when the distribution CD was copied to a network server for network installation. Locate and click the `windows\XeroxFreeFlowVCompose` executable file.
2. Read the Welcome to FreeFlow VI Compose window, then click **Next**. Different levels of software will display different welcome messages.
3. Read the Software License Required window. A new license is required to run this version of VI Compose, and existing licenses will not work. The System Host ID and Product Version are produced on this screen. Record this information, it is required to obtain a new license. Click **Next** to continue with the installation.
4. Read the Software License Agreement window, select **I accept the terms of the license agreement** and then click **Next** to accept the License Agreement and continue with the installation.
5. From the Choose Destination Path for FreeFlow VI Compose window, enter the installation path. Default path is `C:\Program Files (x86)\Xerox\VIPP\xgf` then click **Next**.



Note: DO NOT use the C: drive on the DocuColor Series DFEs. For more information, refer to the *Customer Expectations Document* (CED) supplied with the device.

6. When a previous version of the software has been detected, a Warning screen appears. Select the **OK** option to continue the installation. A backup copy of the xgf directory (xgf\_bak) is created. Select **Back** to enter a different drive, or, select **Cancel** to exit the installation process.
7. When a previous version of the xgfc directory exists, a screen is produced to notify you of the directory's existence. No customer files will be overwritten.
8. From the Select Default Media window, click the button that represents the default paper size to use in VIPP® applications, then click **Next**.

Default media choices are:

- A4: specifies use of A4 size paper (8.25 x 11.75 in. or 210 x 297 mm) as the default media.
  - USLetter: specifies use of US Letter size paper (8.5 x 11 in. or 216 x 279 mm) as the default media.
9. Use the Choose Shortcut Folder window to select the program folder from which to launch VIPP®- related items. Use the default program folder that appears in the text box, or enter a different folder. Once the folder is entered, click **Next**.



**Caution:** Do not select StartUp as the VI Compose program folder.

10. When the Pre-Installation Summary window is displayed, verify the settings specified in the previous steps, then click **Next** to install the software using the settings specified in the previous steps.
11. A window appears showing the installation progress. When the process is complete, a Software license Query window appears.
12. When a license for VI Compose is available, select **Yes**, then **Next**. When no license is available, select **No**.
13. From the license selection window, navigate to the license file, select the license file, and click **Open**.
14. If the license for VI Compose was previously installed or an invalid license was previously attempted for installation, a window containing one of these messages can appear:
  - This file already exists. Replace existing File? Yes/No
  - Access is denied. OK

Click **Yes** and/or **OK** as appropriate.

15. When the Software License Activation Complete window appears, it means that the installation successfully activated the license for VI Compose.  
  
When the Choose Server or File window is produced, the selected license file is invalid. Go to [Installing the License](#) for more information. Another opportunity to specify the license file is provided.
16. The Setup Complete window provides information on the location of the VI Compose release notes so they can be accessed at any time. After clicking **Finish** to complete the installation, a prompt to restart the system is displayed.

This completes the installation. Proceed to [VIPP Golden Job](#) to validate the installation.

### Possible warning and error dialogs

Possible warning or error dialogs are described here.

### Setup Complete With Warning

When the Setup Complete With Warning window is displayed, a license was not provided, or something was wrong

with the license that was provided. In order to obtain a valid license, contact a Xerox representative and provide the Host ID number and the Product Version. Upon receipt of the license, click on **Start > All Programs**, go to the appropriate VI Compose group, then click on **Activate Software License**. The installation does not need to be repeated.



Note: Click on **Start > All Programs > FreeFlow VI Compose x.x > Get Host ID** at any time to display the valid host ID.

### Previous FreeFlow VI Compose Installation Found

When this window is displayed, the system has determined that the installation is an upgrade to an existing VI Compose installation. When this is correct, click **Next** to continue. When this is not correct, or to ensure that no unexpected updates occur, click **Back** to return to the previous window to correct the entries, or click **Cancel** to exit the installation.

### Previous Customer Directory Installation Found

When this window is displayed, the system has assumed that this is an upgrade to a previous version of VI Compose.

When upgrading an existing VI Compose installation, new directories may be added for an existing customer directory, existing files will not be overwritten. Select **Next** to continue the installation.

### VIPP® Golden Job

The VI Compose CD includes many demonstration test files that are automatically loaded in the \xgf\demo directory on the DFE during software installation. Upon completion of the licensing and installation process, the VIPP® golden job should be run in order to validate that the process has been completed successfully.

To submit the VIPP® golden job (goljobv) located in the xgf\demo directory, select **File > Import**. Scroll to `usr\xgf\demo` and select the **goljobv** file.

### Back up VIPP® resource files

In addition to regular backups of the xgf and xgfc directory structures, backups should always be performed upon completion of the installation and prior to reinstalling any software on the DFE, or in the event that installation of new DFE software is required. Performing frequent backups will decrease the likelihood of losing important information in the event of hard drive failure.

Backup is a customer responsibility, and the backup frequency should be based on the backup procedures in place at the customer site.

## LICENSE INSTALLATION

The VI Compose license is installed during the initial device setup process, or when the license is unavailable, it is installed after the initial setup process is complete. When the license is not installed, VI Compose will only run in Demonstration mode. Production mode is activated by obtaining a VI Compose license and installing a device-specific license string.


### Obtain the License String

To obtain a VI Compose license, supply the Host ID and product version to a Xerox representative. The information was displayed on the Software License Required screen when the software was installed. GetHostID.exe can also be invoked to produce the required information. Once the required is obtained information follow these steps:

1. Contact your Xerox representative. Provide the Host ID and product version. The serial number of the printer targeted for installation, and relevant customer account information may also be requested. Specific delivery instructions should be communicated to a Xerox representative.
2. A license file is generated and sent via an email attachment to the email account provided during Step 1.

## Installing the License

To complete the license installation:

1. Upon receipt of the license, save the attachment (.dat file) to an appropriate media to transfer to the DFE.
  2. Determine which license installation process to use. Read the information found in the sections referenced in this bulleted list for more information:
    - **Activation during software installation:** This is the easiest way to install the license.
    - **Simple activation:** Use this process to install the VI Compose license when VI Compose software has been installed but is running in Demonstration mode. It is also the default licensing process during software installation.
    - **Advanced activation:** Use this process to install the VI Compose license in a location other than the default. This option is available during initial installation or when VI Compose is running in Demonstration mode.
-  **Note:** The C: drive on the DFE is the default location for the installation unless an alternate location is specified. To avoid potential space limitations, the Xerox representative should specify a drive other than the C: drive as the destination volume. Xerox recommends installation of the license in the xgf directory.
3. Store the Ethernet card (NIC) number and the VI Compose license file in a safe location for use in the event that the software needs to be reloaded at a future date.

## Activation During Software Installation

The simplest way to activate the license is to have it available before beginning the VI Compose installation; the installation process will prompt you for the location of the license file and do all the work necessary to copy it to the proper location with the proper name, automatically performing the activation process.

When this message appears during the VI Compose installation, the license has already been activated and it is not necessary to read the rest of this chapter:

## Simple Activation

To activate the license after the software is installed:

1. Copy the license file to C:\Program Files (x86)\Xerox\VI PP\xgf\src. If VI Compose is installed in a different location, copy the license file accordingly, to C:\Program Files (x86)\Xerox\VI PP.
2. Rename the license file to license.dat.
3. To activate the license, click **Start**, then select **Programs > Xerox FreeFlow VI Compose > Activate Software License**.

An Activation Complete dialog, which indicates that the license has been activated successfully, is displayed. Now VI PP® jobs can be printed in Production mode.

When VI Compose does not run in Production mode, go to [VI Compose / Makeready API Installation](#) for more

information.

## Advanced Activation

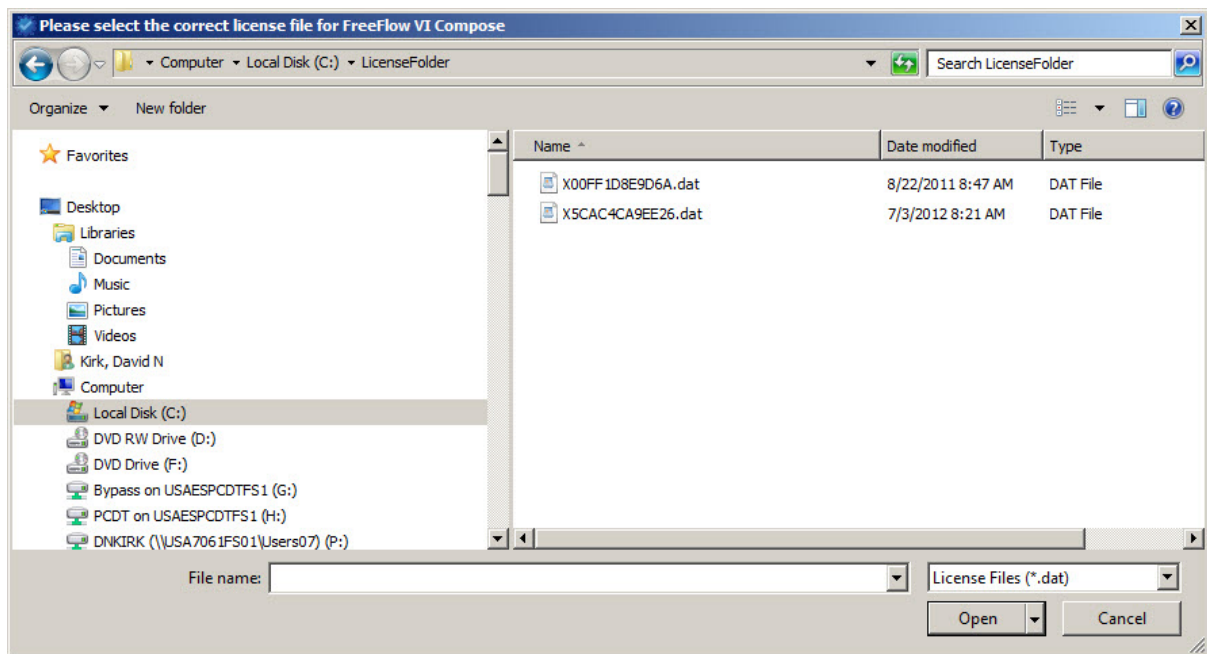
Use this method to activate the license when:

- Your organization keeps its licenses in a central location.
- The license cannot be placed in the C:\Program Files (x86)\Xerox\VIPP\xgf\... directory hierarchy.

1. To begin the license activation process, from the Start menu, click **All Programs > Xerox FreeFlow VI Compose > Activate Software License**.

This dialog appears when no license is found. Contact the license administrator and provide the HostID and version number displayed on this dialog.

2. After you obtain the license file from the license administrator, place it in your system. Browse to the **license file**, highlight it, then select **Open**.



3. Click **Next**.
4. Click **Finish** to complete the license activation process. The license has been successfully activated when this message is displayed:
5. Click **OK** to exit the activation process. If VIPP® jobs can now be printed in Production mode, skip the rest of this chapter. If not, go to [VI Compose Makeready API Installation](#).

## Installing the Makeready Applications Program Interface (API)

In this document, the FreeFlow VI Compose for Makeready is referred to as Makeready. This chapter describes the basic installation of the Makeready API. The Makeready API provides VI Compose the capability to call out RDO files created in the Makeready module.

Installation and configuration of Makeready is described in the *FreeFlow Installation Guide*.

For a description of Makeready and an overview of the VI Compose interface, refer to [VI Compose and FreeFlow Makeready](#).

### Where to install

VI Compose does not have to be installed on the Makeready PC. However, you must install the Makeready API, which is found in the Makeready folder, on the VI Compose.

### File backup

When upgrading or uninstalling this software, be sure to back up any customized files that may be required at a later date.

### Prerequisites for Installing the Makeready API

In order to enable the Makeready API, you must have:

- The VI Compose installation CD
- The Makeready software
- User-supplied Hummingbird NFS mount software or similar

The installation of Makeready involves:

- [Makeready software installation](#)
- [Installing the Makeready Applications Program Interface \(API\)](#)
- [Configuring the Makeready Applications Program Interface \(API\)](#) including mounting the VDI directory on the target print device

## MAKEREADY SOFTWARE INSTALLATION

This is a standard Makeready installation. Refer to the *FreeFlow Installation Guide* for complete instructions on standard installations. All components to support the VI workflow are installed as part of the standard Makeready installation, with the exceptions of the VI Compose for Makeready API, which is described in the following section.

### INSTALLING THE MAKEREADY APPLICATIONS PROGRAM INTERFACE (API)

The VI Compose CD contains a Makeready directory containing the VI Compose and Makeready Applications Program Interface (API) Installation. To install the VI Compose for Makeready API:

1. Insert the VI Compose software CD into the CD drive.
2. Cancel out of the VI Compose installation procedure when it auto-initializes.
3. View the files contained on the CD drive.

4. Open the Makeready folder on the VI Compose CD.
5. Double-click the installation file **setupvdi.exe**.
6. After the installation completes, the PC must be re-booted.

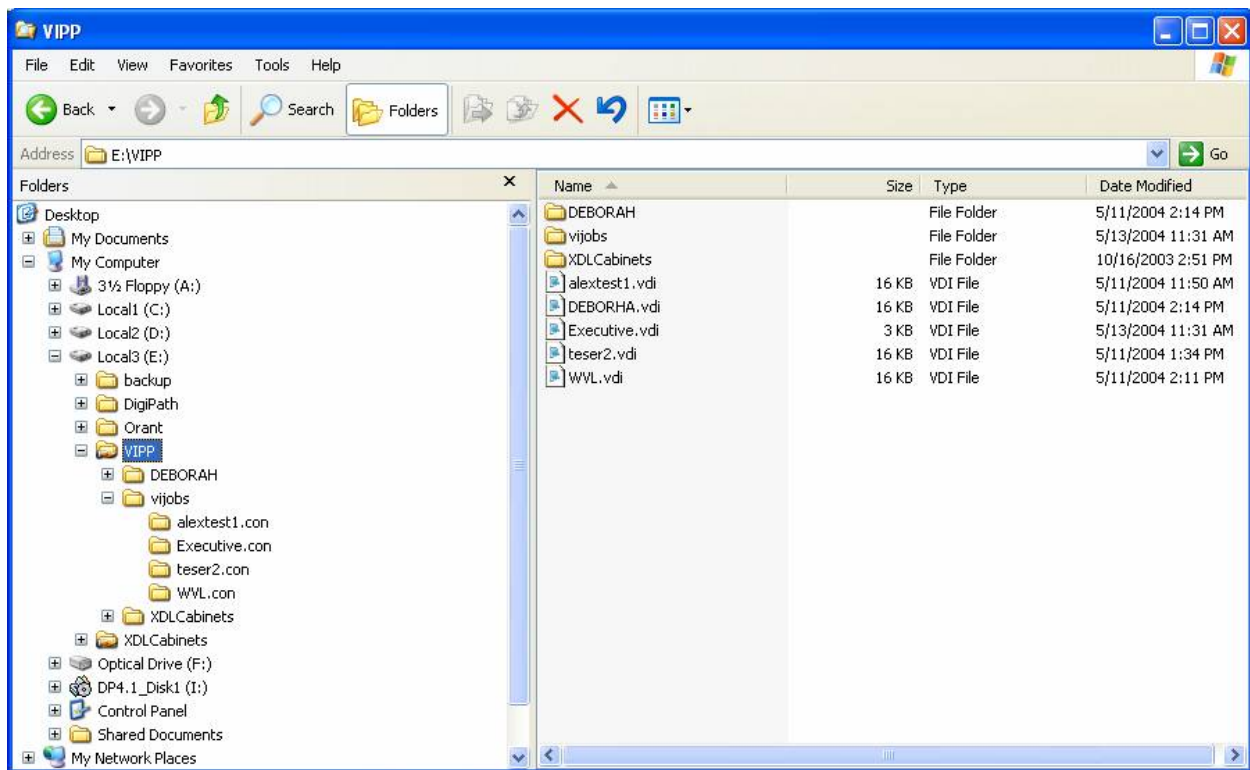
### CONFIGURING THE MAKEREADY APPLICATIONS PROGRAM INTERFACE (API)

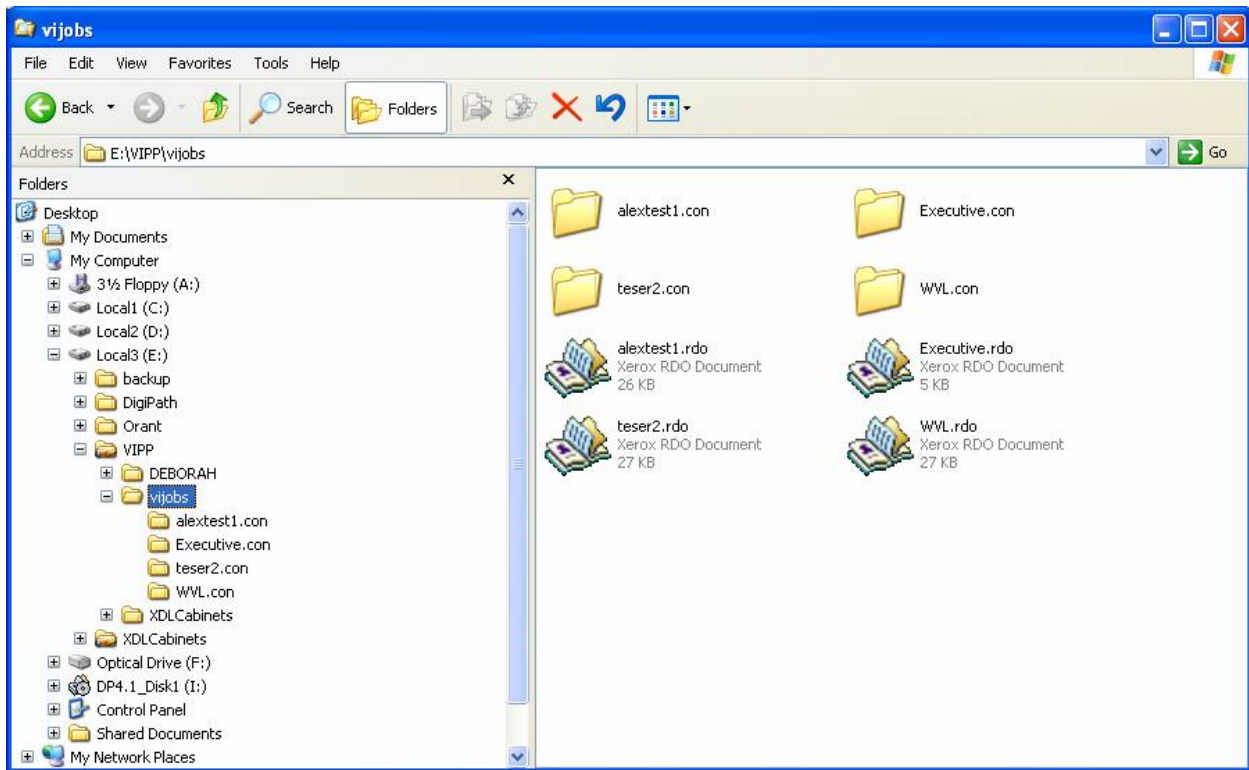
Ensure that when Makeready and VI Compose have been installed on the Makeready PC, that PC and any target print devices, are configured. This involves:

- Makeready configuration
- NFS mounting of the VDI directory
- Identify networked print devices

### Makeready Configuration

The VDI directory represents the mount point for the NFS mounted file system. The screen captures below show the location of a VDI directory named vijobs.





After the VDI directory is shared using Hummingbird NFS or similar software, mount the shared directory at the printer's controller. For UNIX systems like those driving FreeFlow Print Server and DocuPrint, use the mount command. Typically, the mount point is created just under the xgfc directory. Choose any name for the mount point, then create a directory by that name. In our example, the name chosen is: `/usr/xgfc/vijobs`

Last, ensure that the new mount point is included in the libraries defined to VI Compose by the **SETMPATH** command, or by the **SETPPATH** command when using VI Projects, by adding the new mount point to the startup `xgfunix.run` or `xgfdos.run` file. Be aware that UNIX devices may error out when they do not find the mount point readily upon boot up.

When the VIPP® **RUN** command calls out a VDI file, it looks for that file as it would any other miscellaneous file. It searches for the file in each library defined by SETMPATH. In this case, one of the miscellaneous libraries points to the mount point, and the search will be directed to the VDI directory. When VI Compose finds the VDI file, it reads the name of the subdirectory that contains the RDO structure and an index to the TIFF files. VI Compose then processes each TIFF image in succession similarly to the way VI Compose works with Decomposition Service.

## **NFS Mounting of the VDI Directory**

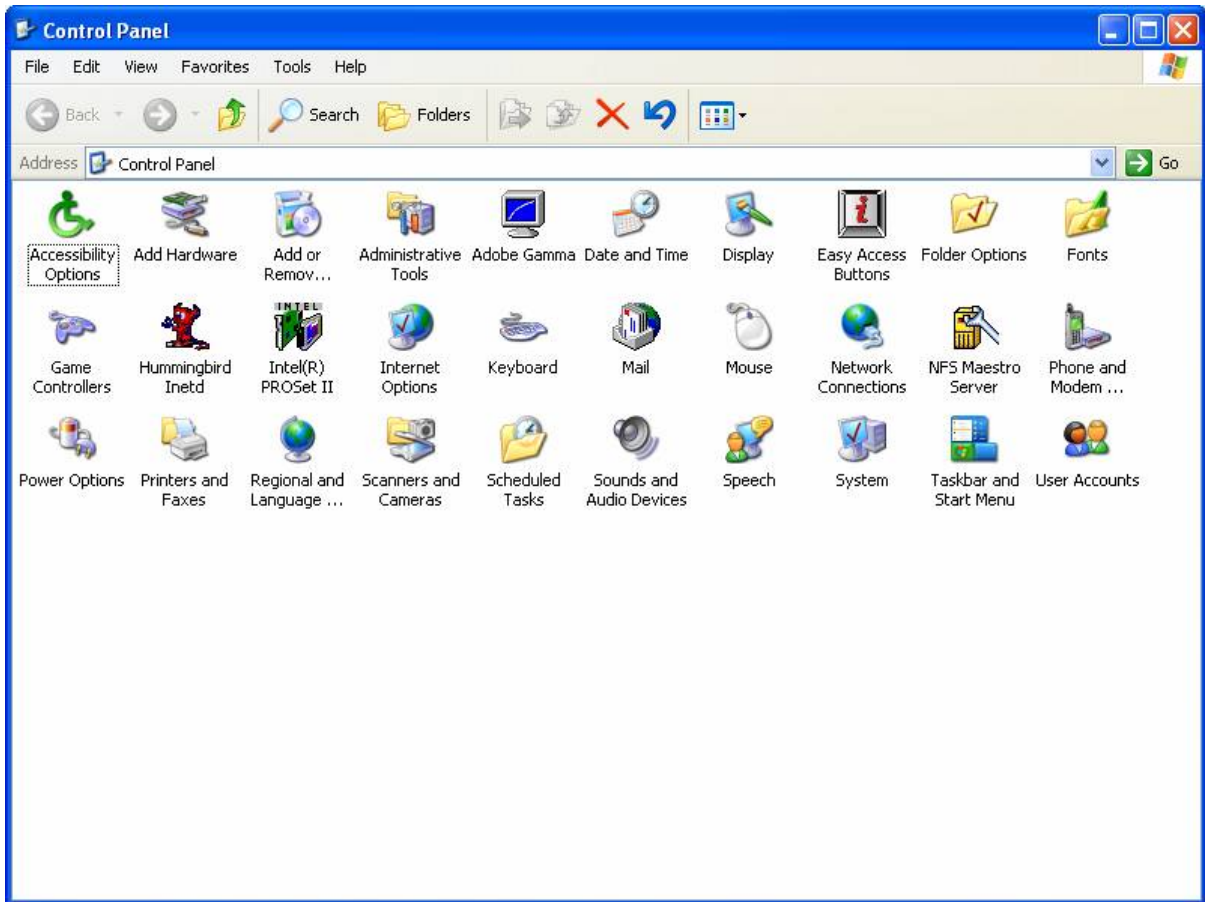
Makeready uses the user-supplied Hummingbird NFS Maestro Server, or similar software, to share, or export, local directories over the network. This software must be installed and configured to export the VDI directory. Also configure the Printer to mount the exported Makeready directory. A VI Compose/ Makeready directory is normally created under the xgfc directory as the mount point.

A network administrator should be involved in setting up the NFS mount and export of the VDI directory to the target print device. Procedures to do this will vary from system to system.

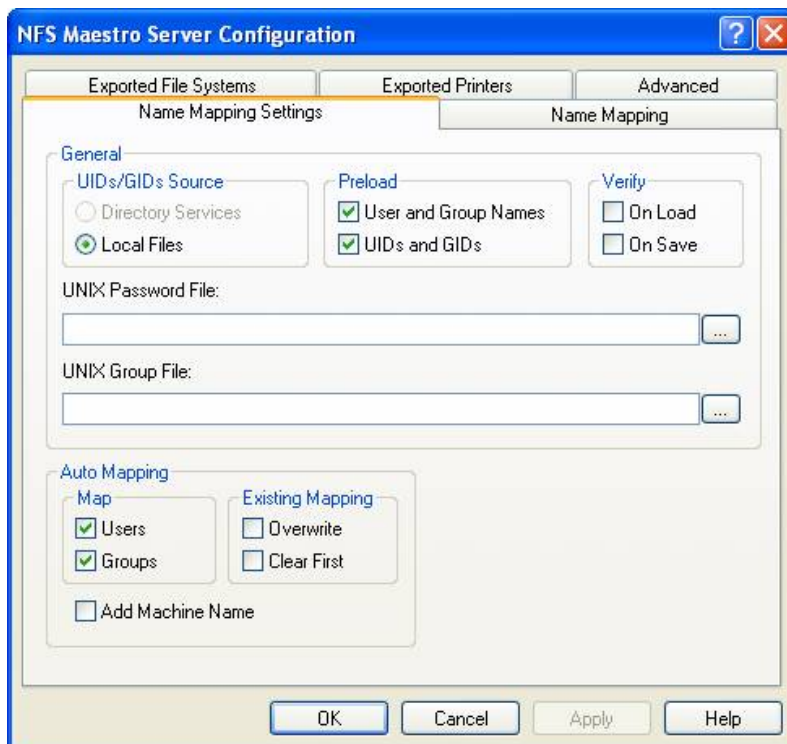
The procedure below should be used as a guide only. It is based on mounting a Decomposition Services directory to a FreeFlow Print Server system. This procedure is covered in the FreeFlow System Administrator Guide for FreeFlow Applications with Document Library, Section 3, Procedure: Configuring the FreeFlow Print Server Decomposition Service. The configuration process may change based on the network configuration and the target device.

The critical steps for Makeready setup are shown here:

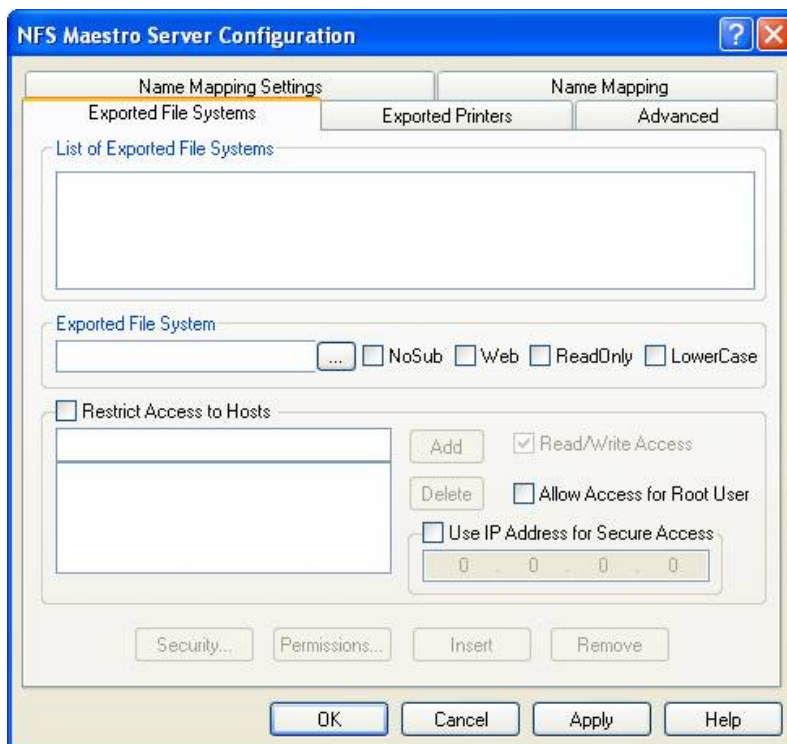
1. Start the Control Panel and choose **NFS Maestro Server**.



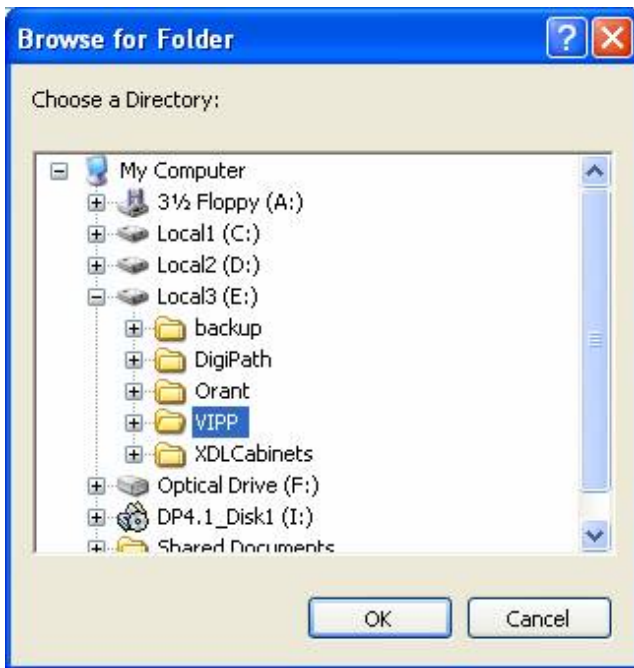
Once you open the NFS Server window, you will no longer see a reference to Hummingbird.



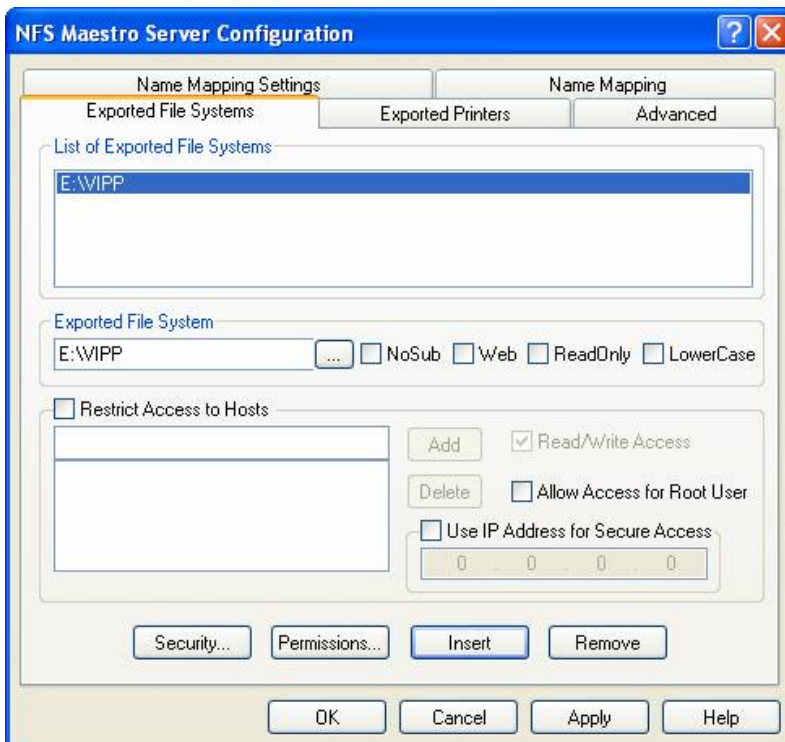
2. Select the **Exported File Systems** tab to view this screen:



3. Select the ... button (Browse) in the Exported File System field. This dialog is produced:

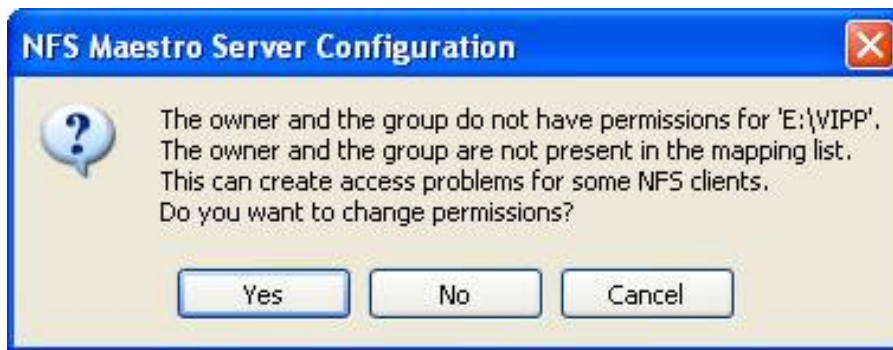


4. Browse to and highlight the **VDI directory**, then press **OK**.



5. When the Exported File Systems window is displayed, ensure that the Restrict Access to Hosts box is **NOT** selected, then press **Insert** followed by **Apply**.

6. When access rights have not been set to Everyone, Hummingbird will prompt you to approve it now. Select **Yes** when you see this dialog:



7. When **Yes** is selected the system produces a Permissions window. Three variations of that screen are shown here, use them as examples to set the appropriate information for the system.

**Permissions**

Current Directory: E:\VIPP

Basic Permissions

	Owner	Group	Other	Read	Write	Execute
	XDL_ADMIN	Users		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
				<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

☐ Recursive

Reload Names OK Cancel

**Permissions**

Current Directory: E:\VIPP

Basic Permissions

	Owner	Group	Other	Read	Write	Execute
	DPAdmin	Users		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
				<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

☐ Recursive

Reload Names OK Cancel

**Permissions**

Current Directory: E:\VIPP

Basic Permissions

	Owner	Group	Other	Read	Write	Execute
	XDL_ADMIN	Users		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
				<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

☐ Recursive

Reload Names OK Cancel

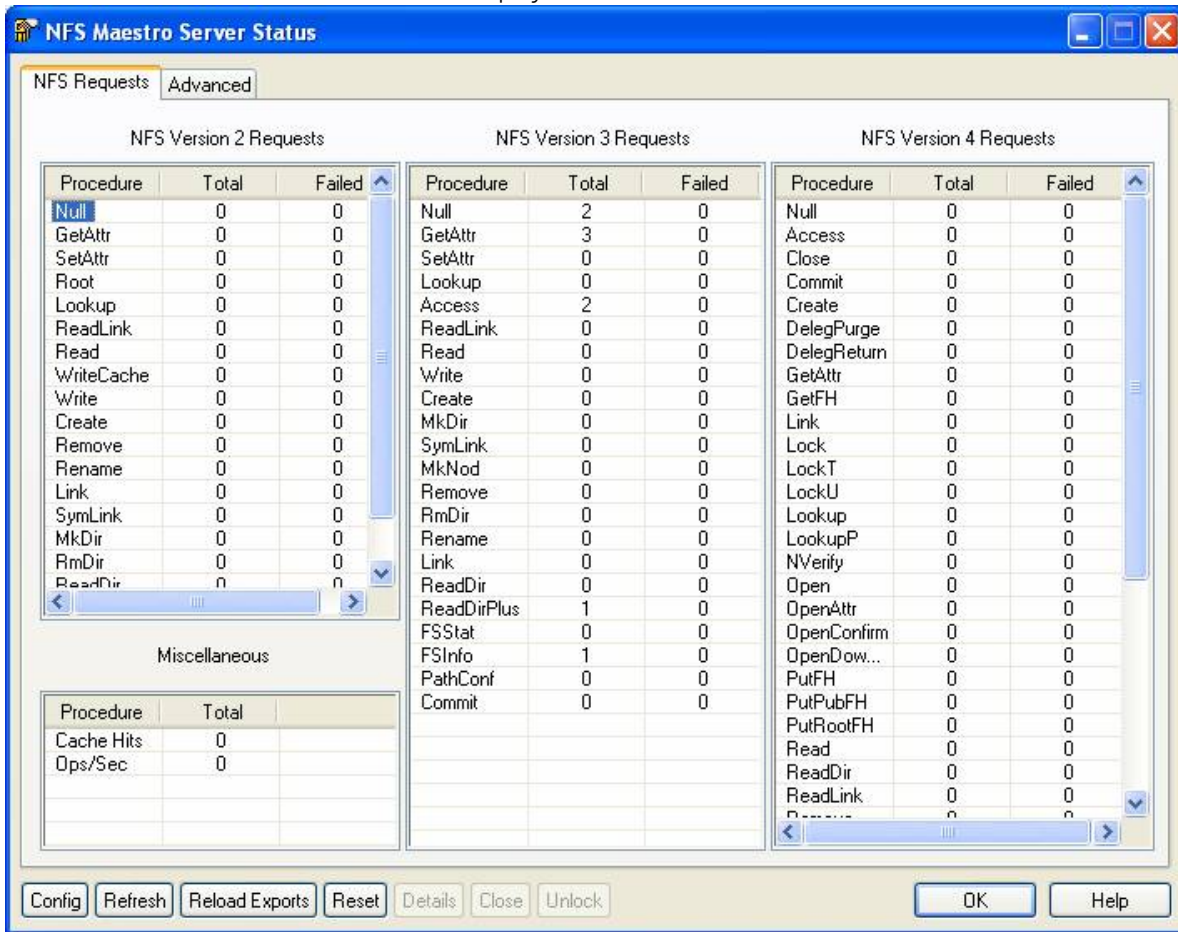
8. After you have set up permissions return to the control panel shown here. Double-click **NFS Maestro Server**.



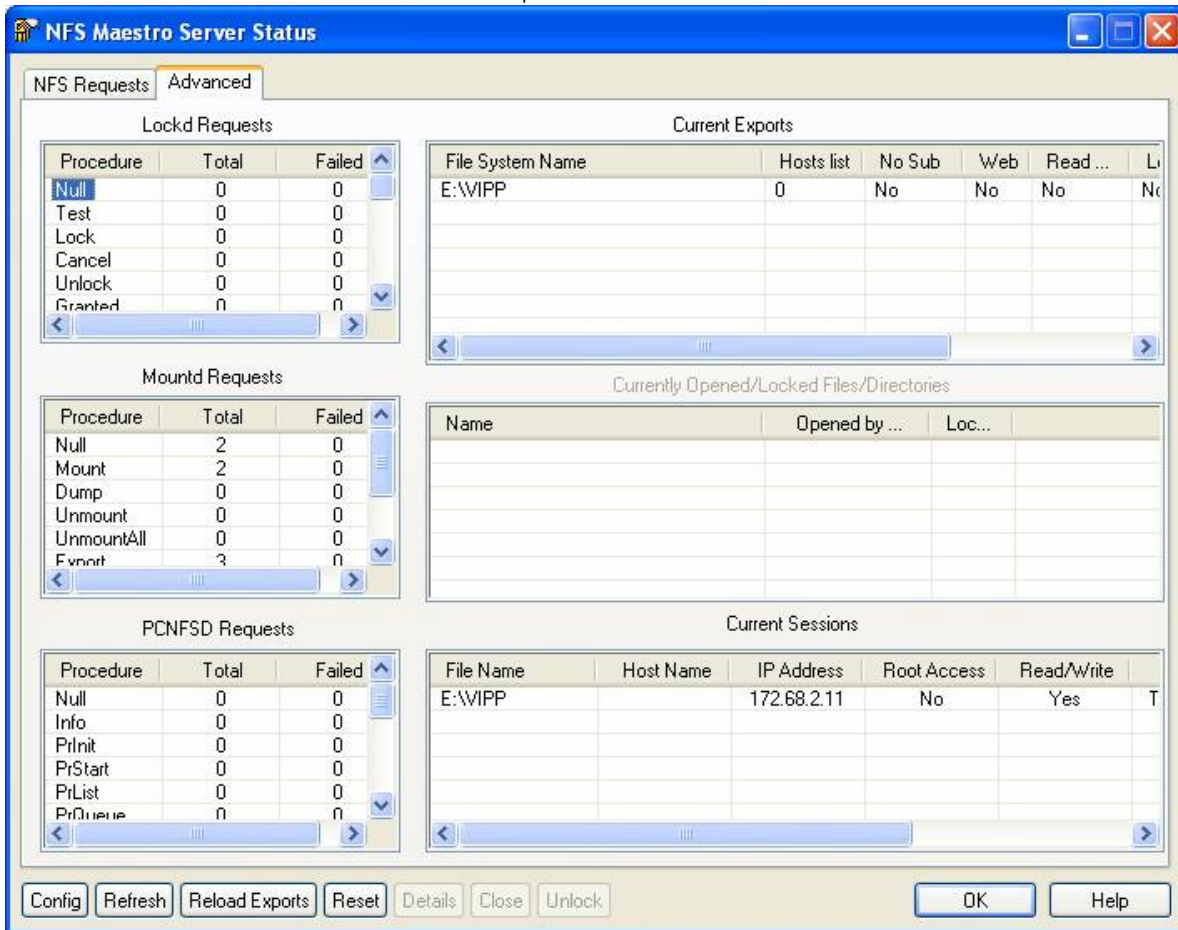
This window is displayed:



9. Double-click **Server Status**. This window is displayed:



10. Select the tab labeled **Advanced**. This screen is produced:



11. Check that the correct VDI directory name appears in the Current Exports window, and click **Reload Exports**.

To set UNIX to re-establish the mount point every time it is rebooted, get the UNIX system administrator to edit the `/etc/vfstab` file.



**Caution:** The syntax of the `/etc/vfstab` file may change, depending on the operating system and/or version in use. Always validate the correct syntax for the system before making any modifications. The examples shown are provided for a Solaris 8 system.

Add an entry along the lines of:

```
13.245.64.152:/G/Product_Z/Marketing - /usr/xgfc/VDI_dir nfs -
yes bg, rw, hard, retry=5
```

Notice that "-" is entered to prevent UNIX from attempting to perform a file system check (fsck) on the volume. Also ensure that `bg` is specified in the option string so that UNIX knows it can skip the mount point when the VDI volume is not available when you boot the printer controller. This entry tells UNIX to attempt to reconnect to the volume in the background when it cannot find the connection on the first attempt. Last, specify a hard link when you connect to the volume with `rw` (or read-write) privileges.

## Identify Networked Print Devices

The VI Compose installation on your print device needs the correct path to the resource directory.. Failure to properly link the software to the resource directory results in this error message and the VIPP® job aborts.

VIPP\_no\_valid\_path\_or\_access\_denied

The VIPP® programmer should work closely with the system administrator to ensure that:

- All Makeready resource files are located in the Makeready VDI directory
- The correct mount point has been identified at the print device
- The xgfunix.run file has been modified to point to that directory

The system administrator must create a directory under the xgfc directory on the print device to match the name of the NFS mount point. For example:

```
/usr/xgfc/VDI_dir
```

## Update mislib

The final step in the installation process is to update VI Compose list of miscellaneous libraries (mislib) by updating the VI Compose Initialization files, xgfunix.run or xgfdos.run.

The xgfunix.run file is located in the /usr/xgf/src directory on all UNIX based systems. Look for the SETMPATH command and make the changes shown in bold text.

```
[ (/usr/xgfc/mislib/)      % customer path for miscell. files
  (/usr/xgf/demo/)        % demo path for demo files
  (/usr/xgfc/formlib/)    % customer path for forms & segments
  (/usr/xgf/formlib/)     % demo path for forms & segments
  (/usr/xgf/encoding/)    % demo path for encoding files
  (/usr/xgfc/VDI_dir/)    % path for VDI files
  ()                      % null path for full names
] SETMPATH
```

On Windows-based systems, edit xgfdos.run, which is usually found in C:\Program Files (x86)\Xerox\VIPP\xgf\src. The line you enter is shown below. If you have installed the program to a path other than C:\Program Files (x86)\Xerox\VIPP, use the path designation for the xgfc directory.

```
(C:\Program Files (x86)\Xerox\VIPPC\xgfc\VDI_dir\ ) % path for VDI files
```

When you are using VI Projects, also add this line to the list of directories defined by SETPPATH.



### Note: Controller reboot

Any changes to VIPP® configuration files require a reboot of the FFPS controller.

## VI Compose Installation on DocuPrint NPS Devices

Use these instructions to upgrade VI Compose software and install the VI Compose license on all DocuPrint NPS devices including DocuPrint NPS color devices.

VI Compose software is installed in Demonstration mode during the normal DocuPrint NPS software installation process. There are no special installation instructions for the initial installation of the VI Compose software. However, in order to run the software in Production mode a VI Compose license must be installed and activated.

When VI Compose has already been installed, these two directories will exist on the system:

- `/usr/xgf`
- `/usr/xgfc`

### CUSTOMIZED FILE BACKUP

During an upgrade, customized VI Compose files located in `/usr/xgf/src` and `/usr/xgf/encoding` are overwritten with new files of the same name. The system automatically copies and renames those directories as a form of backup for customized files. The backup versions of the original directories are `/usr/xgf/src.$OLDVER` and `/usr/xgf/encoding.$OLDVER`, where `$OLDVER` represents the previous version of VI Compose. When customized files exist in either of the original directories, it is your responsibility to merge the customized contents of those files with the new versions found in the most recent `/usr/xgf/src/` and `/usr/xgf/encoding` directories.

Typically, customization is done in one of the following files. Even though the file contents will be backed up during an upgrade installation, Xerox recommends that the files be printed as hardcopy backup.

<code>xgf/src/xgfunix.run</code>	VI Compose startup definition file
<code>xgf/src/xgf.pcc</code>	standard pcc definitions
<code>xgf/src/xgf.mup</code>	Multi-Up definitions
<code>xgf/src/xgf.gep</code>	color and graphical element property (gep) keys
<code>xgf/src/xgf.bat</code>	background attribute (bat) keys
<code>xgf/src/xgf.def</code>	VI Compose system defaults
<code>xgf/encoding/fontlist</code>	(found in <code>xgf\encoding</code> )



Note: Any previous customization of the VI Compose environment will not be valid in a new upgrade. Copying the old files has the potential of rendering the new functionality unusable. Any customization will have to be repeated on the new files after a successful upgrade.

## SOFTWARE INSTALLATION

An upgrade may be necessary to load the latest version of the VI Compose software. Upgrading VI Compose software from the CD will not overwrite the customer files located in `/usr/xgfc`, however it will overwrite the `/usr/xgf` directories.

Performing an upgrade maintains the validity of an installed license file.

To upgrade VI Compose on a DocuPrint NPS controller:

1. Log on the system at the "root user" security level.
2. Locate the existing VIPP® package name(s), enter:

```
pkginfo -c VIPP | grep VIPP
```

```
pkginfo | grep XRXdpxgf
```

VIPP® is the package category of the VI Compose software as it is loaded as an upgrade. XRXdpxgf is the package name of the VI Compose software as it is loaded with NPS software.



Note: Do not remove the package named XRXdpvip as this is the Xerox DocuPrint Licensing package. When the package is removed, the Show License Status command will report "Could not get any license status" even though the license files are present and active. To correct this problem, the NPS software will have to be reinstalled.

When more than one VIPP® package name is displayed (other than XRXdpvip, they must all be removed; use a last installed to first installed order when removing them.

3. Remove the existing package(s):

```
pkgrm package name
```

**Where:**

**package name** is the result from Step 2. For example: `pkgrm VIPxx`

4. Log on as administrator in the NPS/IPS Command Line administrator window. Enter: **Privilege Administrator Password** (Contact the system administrator for the correct password.)
5. Begin the software installation. Enter:

### Install Update From CDRom

Enter the UNIX root password when prompted. Contact the system administrator for the correct password.

6. When the VIPP® files were previously customized enter those changes in the new files now. Refer to [Customized file backup](#).
7. Validate the software installation by printing `/usr/xgf/demo/go1job`. Place color stock in trays 1, 2, and 3 as follows:
  - Tray 1 - Yellow
  - Tray 2 - Green
  - Tray 3 - White

The completed job consists of 43 single-sided pages. Pages 1 and 43 are printed on yellow stock, pages 8, 10, 25, 26, and 27 are printed on green stock and the remaining pages are printed on white stock.

## LICENSE INSTALLATION

1. Determine if a VI Compose license has been installed. At the UNIX workstation, enter:

```
pkginfo | grep VIP
```

When XRXdpvip is included on the list of packages a license has been installed and no further action is necessary. When not included, proceed to the next step.

2. Acquire the license, refer to [Obtain a VI Compose License](#) for more information.
3. After receipt of the license string, follow one of these two options to complete the license installation:

### Install the License from Faxed Information

To install the license from faxed information:

Log on as Privilege Administrator and at the DocuPrint NPS user interface window, enter: **Install Feature License**

When prompted, enter the information received from a Xerox representative. When complete, a message similar to this example is displayed:

```
name> XRX_VIPP_software
expiration date>01-Jan-1999
key> 1EF830EABF32453663D8
print engine serial number> (Leave blank for VIPP license)
Hostid> xxxxxxxx
```

### Install the License from a Disk or Email

To Install the license from a disk or email:

Copy license.dat from the email to a floppy disk. Insert the floppy disk into the floppy drive of the NPS controller and at the user interface enter:

#### Install FlexLicense From Floppy

At the prompt, "Floppy disk inserted? (no to quit)" enter:

**Y**

The system will report the number of licenses that have been pre-pended to the license file. Then, when prompted for another disk, enter:

**N**

Finally, at the "Floppy disk inserted? (no to quit)" prompt, enter:

**N**

## Validate License Installation

1. To validate the license installation, log on as administrator in the NPS/IPS Command Line administrator window and enter:

### Show License Status

This is an example of the results of the command:

```
XXR_VIPP_Software License will expire in xxx days.
```

### Where

**xxx** is the number of days that the license is valid.

2. Restart the NPS sequencer. At the command prompt, enter:

### Restart Sequencer

## License Problem Solving

Perform these actions in sequence when the VI Compose license string is not accepted by **Install Feature License** or **Install FlexLicense From Floppy**, or when after a successful installation of the VI Compose license string, Production mode is not activated:

- Use Show License Status to check for the existence of installed licenses.
- Re-enter the VI Compose license string exactly as delivered by a Xerox representative. Remember that the license string is case sensitive; enter it exactly as shown. Contact a Xerox representative when there is a question about a character.
- Check that the correct VI Compose license string has been installed in the corresponding printer. Remember that the VI Compose license string is printer specific. Use the Show HostID command to display the Host ID and compare it with the one in the fax sent by a Xerox representative.
- Check that the Host ID entered in the VI Compose License Request Form is correct. When it is not, contact a Xerox representative.
- Contact a Xerox representative to confirm the accuracy of the VI Compose license string they provided.
- Contact the Xerox Help Line.



# Variable Information Suite Documentation

This chapter contains:

Intended Audience..... 48

Documentation Overview ..... 49

The *FreeFlow® VI Compose User Guide* provides background information on FreeFlow Variable Information Compose software, its use of VIPP® files, resources and datastreams, and documentation for the FreeFlow Variable Information Suite (VIS).

This chapter includes these sections:

- Intended audience
- Documentation overview

## Intended Audience

FreeFlow Variable Information Suite Documentation is intended for:

- Application programmers who want to produce a VIPP® data stream for native mode users
- Layout designers who want to use VIPP® to encode the layout definitions for line, database, and XML mode users
- Developers of advanced layout design applications who want to include VIPP® as a format for the resources produced by their products for example, VIPP® emitters
- System administrators responsible for system configuration, security, and license information



Note: When using VI Compose or any of the Variable Information Suite of applications, it is assumed that you have experience using the VIPP® programming language. It is also assumed that you have a basic knowledge of the VIPP® programming conventions.

It is also assumed that you have a working knowledge of the computer and its operating conventions, including how to:

- use a mouse and standard menus and commands
- open, save, and close files

## Documentation Overview

English language versions of the *FreeFlow VI Compose User Guide* and all other Variable Information Suite documentation can be downloaded from [www.xerox.com/support](http://www.xerox.com/support) and refer to **Program download**. When requested to click on **Software**, click on **Documentation** instead.

FreeFlow Variable Information Suite Documentation includes these books:

### **What's New**

This book provides an overview of new features, enhancements and software fixes since the last release.

### **FreeFlow VI Compose User Guide**

This book provides the background information required to understand and use VI Compose and its applications. It describes the files and utilities provided with the software, the resources necessary to build both legacy jobs and VI Projects, and the basics of printing VIPP® jobs using VI Compose.

### **FreeFlow VI Compose (Open Edition) Installation and Overview**

This book provides instructions for installing VI Compose on non-Xerox printing devices.

### **VIPP® Language Reference Manual**

This book documents the VIPP® commands, VI Compose error messages, and programming tips.

### **FreeFlow VI Design Pro User Guide**

This book contains information on the features and functions provided by VI Design Pro, as well as how to use these capabilities to access, view, modify, create, and execute VIPP® applications.

### **FreeFlow VI Design Express User Guide**

This book contains information on the features and functions provided by VI Design Express, as well as how to use these capabilities to access, view, modify, create, and execute VIPP® applications.

### **FreeFlow VI eCompose User Guide**

This book contains information on using VI eCompose to create and dispatch Adobe PDF documents, and to remotely administer VI eCompose Web servers.

### **FreeFlow VI Explorer User Guide**

This book contains information about how to use VI Explorer, a GUI program that allows users to preview and verify VI jobs and projects, and provides easy access to Normalization and Demographics services.

### **VIPP® Manage User Guide**

This book provides information on the use of the VIPP® Manage utility, which is used with VIPP®-enabled desktop or office print devices to load VI Compose software and production license, install VI Compose patch files, and manage VI resources.

### **FreeFlow VI eCompose Dispatch SDK**

This Software Developer's Kit provides information about VI eCompose Dispatch Module functionality.

### **FreeFlow VI eCompose Workshop**

This book contains a tutorial designed to familiarize you with the operation and use of VI eCompose.

## **FreeFlow Variable Information Suite Glossary and Quick Reference**

The glossary contains a list of the terms and acronyms needed to use VI Compose and its applications. The quick reference is a printable version of the Quick Reference Card, which contains a list of all VIPP® commands and variables sorted by function and variables by type.

## **FreeFlow VI SAP Device Type Installation and User Guide**

This book describes how to install and use the Xerox VIC Device type, form, and test job in SAP.



Note: For information about VI Compose training contact a Xerox representative.

# VIPP® and VI Compose Overview

This chapter contains:

Basic Concepts .....	52
SAP Device Type for VI Compose .....	57
Using PDF Resources with VIPP® APPE, PDF/VT, and Embed EPS Options .....	58
VIPP® and Variable Information Suite Applications .....	65
FreeFlow® VI Compose Open Edition .....	72

This User Guide is designed to provide information about how to use a properly installed version of VI Compose. Refer to [FreeFlow VI Compose Licensing and Installation](#) for installation instructions.

Variable Information Production PrintWare (VIPP®) is a variable document composition language that provides nearly unlimited capability and flexibility to its users. VIPP® is dedicated to the design of variable information (VI) applications. VIPP® can be used to describe both the structure and logic of the variable data to process, as well as the appearance of the pages to be produced.

Xerox has implemented a VIPP® interpreter, VI Compose (VI Compose), as a set of high level functions written with the PostScript language. VI Compose adds to PostScript functionality and enhances the ability to print complicated documents at the rated speed of the print device. This means that the job containing variable data, images, logos, and other memory-intensive job segments can be printed without the delays and large intermediary files that are encountered in applications similar to VI Compose.

The *VIPP® and VI Compose program overview* provides the background information you need to consider when using the VIPP® language and VI Compose to print the jobs. It also describes the Variable Information Suite applications that add functionality and make VIPP® easier to use.

These chapters describe VIPP® and VI Compose functionality in more detail:

VIPP® data streams	Describes the modes in which VIPP® jobs are created
VIPP® resources	Describes resources used to create the VIPP® jobs
VI Compos e files and utilities	Lists the files and utilities provided with VIPP®
Standard lists, tables, keys, and attributes	
VIPP® file examples	
Printing with VI Compose	

## Basic Concepts

VI Compose is PostScript software that sits on the printer and allows you to produce complex documents and print them at production speeds. VI Compose features and functions provide enhanced variable data printing performance through the use of cached object elements, dynamic text flow, data driven graphics, data formatting, conditional processing, and workflow improvements. Those elements are all described in greater detail in other sections of the VI Compose documentation set.

In order to use VIPP® and VI Compose effectively, you need to understand these basic concepts:

- Ensure that **VIPP and PostScript** are not mixed
- What **VIPP-enabled devices** are
- What **Repositories** are
- What **VI Projects** are

### VIPP® AND POSTSCRIPT

The first thing to remember about using the VIPP® language is ensuring that VIPP® commands and PostScript is not mixed in the programs. The only time PostScript code can be safely used is in self-contained resources (such as EPS files) that will be used as forms or segments in a VIPP® job or VI Project.

The technical reason for not mixing VIPP® and PostScript is that the VIPP® language uses a postfix notation syntax similar to PostScript. VI Compose has been implemented using the PostScript language and is required to run inside a PostScript interpreter. Because of this situation, intermixing VIPP® and PostScript code may happen to work in some circumstances. However, this kind of mixed code is not compliant with the VIPP® language specifications and will break any application or utility expecting a compliant VIPP® data stream. There is no assurance that such jobs will work correctly with future VI Compose releases. For this reason, mixed code is not supported and must be absolutely avoided.

### VIPP®-ENABLED DEVICES

The term VIPP®-enabled is used throughout the documentation to describe the devices that support the VIPP® language. The obvious application for VI Compose is as device-resident software on a Xerox printer, however, VI Compose can also be used on these VIPP®-supported devices:

- DocuPrint NPS (monochrome and color)
- FreeFlow Print Server
- External EFI and controllers
- FreeFlow MakeReady
- Limited Xerox Office Devices

#### **Not all devices are supported in all countries**

Contact a local Xerox representative for an updated list of supported printers.

#### **Limited support**

Limited support is provided for Xerox Office Devices. Validate VIPP® operability prior to purchase in an Office device for VIPP® enablement.

VI Compose can also be used to run VIPP® either permanently or on a job-by-job basis on a VIPP® imager (PostScript-enabled printer, fax machine, computer screen, etc.). To use VIPP® capabilities with a VIPP® imager:

- Embed VIPP® commands in the print data
- Describe the document layout in a Job Descriptor Ticket (JDT), XML Job Ticket (XJT), or a Data Base Master (DBM), which you then make available to the VIPP® imager

Whenever possible, use supported devices that run a PostScript interpreter with access to a local file system, for example, a DocuPrint NPS, FreeFlow Print Server, or a PC or UNIX workstation. These devices access VIPP® resources in shared libraries on that file system. In addition, other file systems can be NFS mounted to provide additional resource libraries.

Desktop VIPP®-enabled printers, such as N-Series, DocumentCentre, or Phaser printers, can also access VIPP® and VI Compose on a local file system. VIPP® Manage is used to control, download, delete, and print VIPP® resources to the printer disk, or to send self-contained VIPP® jobs to diskless printers.

When devices with access to a local file system are not available, use supported devices that run a PostScript interpreter without a file system, for example, a diskless DocuPrint N40. These devices must embed resources in the data stream before a file is submitted to the printer.

## REPOSITORIES

A repository is a collection of VIPP® resources and projects that is associated with a single xgfdos.run (Windows) or xgfunix.run (FreeFlow Print Server) file. On FreeFlow Print Server, there is a single xgfunix.run file, and thus a single repository. On Windows there can be multiple xgfdos.run files, one for VI Compose, one for VI Design Pro, one for VI eCompose, and others for possible user-defined repositories.

The xgfdos.run and xgfunix.run files contain VIPP® commands that determine the VIPP® resource directories. For VI Projects, the command is **SETPPATH** which is used in conjunction with **SETPROJECT**, refer to [Project directory structure](#). For non-project (legacy) resources, the commands are **SETEPATH**, **SETFPATH**, **SETIPATH**, **SETJPATH**, and **SETMPATH**.

## VI PROJECTS

VI Projects provide a mechanism that enables Variable Information Suite applications to manage all the resources of a job as a single entity. VIPP® job resources that are organized and stored in libraries by project can be manipulated as a single VIPP® job. Handling job resources is simplified by logically and physically grouping them as part of a VI Project, which is the logical grouping of the physical resources used by the job at one or more steps in the job life cycle.

VI Projects allow you to identify, organize, and store the resources of a job under a single name (the project) and group the jobs by family (the folder). A folder is a collection of projects that share some common features. For example, a Service Bureau creates one folder per customer, and stores each of that customer's jobs as a project under that folder. This is a sample structure:

Customer Folder A

Job 1

Job 2

Job 3

Customer Folder B

Job 1

Job 2

Job 3

Customer Folder C

Job 1

Job 2

Job 3

The physical components of a VIPP® job consist of resources such as:

- Pictures in TIFF, EPS, or JPEG format
- Job Descriptor Tickets (JDT) or Data Base Masters (DBM)
- XML Job Tickets (XJT)
- Forms or segments in VIPP® or PostScript format
- Fonts
- Encoding tables
- Submission files
- Production data
- Editable versions of some resources in application-specific format
- Viewable versions of the job in Adobe PDF format
- Pre-processed objects stored in optimized application-specific format
- **VI Project File (VPF)**

For additional information, refer to [VIPP resources](#).

### **VI Project File (VPF)**

A VI Project File contains all the information needed for a single project. A VPF always has a .vpf extension, and is stored in its project's directory (which also contains the rest of the project's local resources, refer to [Project resource scope](#) and [Project directory structure](#)).

Applications handling VI Projects need to know more than the location of the components on the file system in order to provide high-level functionality. For example, because VIPP® does not enforce naming conventions for the files containing the project's components, the type of the named component (form, segment, etc.) must be provided in the VPF.

The VPF contains four kinds of information about each project, which are stored as XML elements or attributes:

- Bibliographic information
- VIPP® resources
- Non-VIPP® resources
- Historical information



Note: A .vpf file will be contained in VI Projects created by FreeFlow VI Design, or by software provided by a third party who has implemented the VI Projects SDK. Projects created by a third party who has not implemented the VI Projects SDK, will not have a .vpf file, and are not supported by Xerox.

## VI Project Containers

A VI Project Container (VPC or .vpc) is a compressed archive file containing the physical grouping of all the VI Project's resources, including the VPF. VPCs always have a file extension of .vpc. VPCs simplify the transferring of projects between applications, devices, or locations.

The FFPS has the ability to accept a VI Container (VPC) as a print file. By enabling the VPCF filter (found under the FFPS Queue menu) a VPC file can be deployed, a VPC file can be deployed and printed, or you can print and forget a VPC file. These options will expand a VI Container file, then, depending upon the selections you make, can print the data file and can delete all files once the job is printed.

## Project resource scope

Each project resource has one of the following scopes:

- Local
- Folder-shared
- Global-shared

Local resources are resources that can only be used by the project to which they pertain. Local resources are stored in the project's directory (refer to [Project directory structure](#)).

Folder-shared resources are resources that can be shared among all projects in a specific folder. A logo shared across all projects within a folder assigned to a specific department can be assigned a folder-shared scope. Folder-shared resources are stored in one of the folder-shared directories within the folder to which they pertain (refer to [Project directory structure](#)).

Global-shared resources are resources that can be shared among all projects in all folders accessible by the application. A company logo may be shared across all the projects within a company. In this case it makes sense to avoid storing the logo in every folder or project. Global-shared resources are stored in one of the repository's global-shared directories (refer to [Project directory structure](#)).

Local resources take precedence over folder-shared and global-shared resources, i.e. when a resource in local scope is also available with the same name in either of the other two scopes, the local resource will be used. Also, folder-shared resources take precedence over global-shared resources.

Sharing resources at folder-shared or global-shared scope saves storage space, ensures consistency, and simplifies maintenance.

## Project directory structure

Following is an example of a set of project resource directories for a single repository. Note that the names of the directories are arbitrary, and can be any valid file names.



The names of the global-shared, folder parent, and folder-shared directories (in red) are determined by the value of the VIPP® command SEPPTATH, which is defined in the file xgfdos.run (Windows) or xgfunix.run file (UNIX) for the repository. The global-shared and folder parent directories do not have to be in the same directory; they can be anywhere in the file system, and their locations are independent of one another. In fact, a global-shared directory can be a subdirectory of a folder parent, as in the default structure shown below.

Within one folder parent, the names of the possible folder-shared directories in each of the folders are the same, although a folder does not need to have all of the possible folder-shared directories allowable for a folder parent. Folder2 above illustrates this. Other folder parents may have different names for the folder-shared directories.

The names of the folders and projects (in blue) are determined by the value of the VIPP® command SETPROJECT, which must appear in each project-based VIPP® job, and gives the folder and project names for the job. The VPF file for a project also stores the folder and project names for the project.

The folder names must be unique, i.e. two folder parents cannot each have a subdirectory with the same folder name. However, two folders can contain projects with the same name. The uniqueness of the folder names ensures that a given combination of folder name and project name identifies at most one project.

During the first installation, VIPP® is configured by default with the following VI Projects repository:



The root path for xgfc will be either /usr/xgfc on UNIX systems, X:\Program Files (x86)\Xerox\VIIPP\xgfc, where X is the partition where the VI eCompose software is installed and %XPS\_DATA%\data\xgfc on Windows FFPS.

## Creating and editing VI Projects

VI Design Pro can be used to edit and create VI Projects. For more information on this tool, refer to the *FreeFlow VI Design Pro User Guide*.

## SAP Device Type for VI Compose

The Xerox FreeFlow VI SAP Systems, Applications and Products in Data Processing Device Type and Form provide functionality and feature integration with the FreeFlow VI Compose and VI eCompose products. Although SAP provides support for many output devices, this support does not include the VI enabled printers and VI enabled eCompose Servers (VIPP® to PDF).

The FreeFlow VI SAP Solution allows SAP users to access the more advanced feature set of the VIPP® scripting language, providing a fast, efficient, effective, and flexible method for printing (or creating Adobe PDF files) from a delimited (XML or line data (print ready data)) data file.

The sample device type and form are provided in a zip file (VISAPDT.ZIP) that can be downloaded from the [www.xerox.com](http://www.xerox.com) site. Go to the Support & Drivers section and search for VIPP®, then select the Software option. The zip file will be found under the Utilities & Applications section. The contents of the zip file should be reviewed by the SAP systems administrator.

SAP will use the legacy VIPP® path using the Just Send the Data model. VIPP® resources, fonts, images, forms, and so on, must be uploaded to the printer's XGFC folders, which can be done during the development of the VIPP® application. Once the resources are available to the print device, or the FreeFlow VI eCompose server, the SAP print job can be processed using the SAP Device Type that has been configured with the VIPP® start commands. This is, in effect, just sending the data to the target device.

The only requirement is that FreeFlow VI Compose is installed and licensed on the target print device. If the SAP administrator needs to output PDF, those VIPP® jobs can be sent to the FreeFlow VI eCompose server or the VI eCompose Web Job Submission Service (WJSS).

## Using PDF Resources with VIPP® APPE, PDF/VT, and Embed EPS Options

VI Compose can support applications with PDF resources in the legacy PostScript RIP, in the Adobe PDF Print Engine (APPE) RIP, and in the various PDF output capabilities of FreeFlow VI Suite (VI eCompose and PDF Export functions). However, the process to enable the use of PDF resources, and the structure and output of those PDF resources differ depending on the intended RIP or target for applications that use them.

### PDF as a resource

Summary of print / display scenarios				
VIPP job displayed or printed on	PDF resource		Output	
FFPS/APPE	NR		PDF	
PS printer VI Design Pro VI Explorer	with EPS		EPS	
	w/o EPS		Text Pattern (max. 10 pages on printer)	

PDF output displayed or printed on	PDF resource	PDF resource embedded	PDF resource external present	PDF resource external not present
Acrobat (>10.1.6 + privileged folder) or APPE	with EPS	PDF	PDF	EPS
	w/o EPS	PDF	PDF	Text Pattern
Acrobat (<10.1.6 or not privileged folder) or XObject not supported (PS rip)	with EPS	EPS	EPS	EPS
	w/o EPS	Text Pattern	Text Pattern	Text Pattern

VI Design Express (Adobe InDesign plug-in) and VI Design Pro tools support using PDF resources in the creation of VIPP® applications. However, the process to use and display PDF resources in these tools is again, determined by the intended target for the VIPP® application.

The purpose of this section is to describe how to use PDF resources in VIPP® applications for the targeted RIPs mentioned above. The following sections are included:

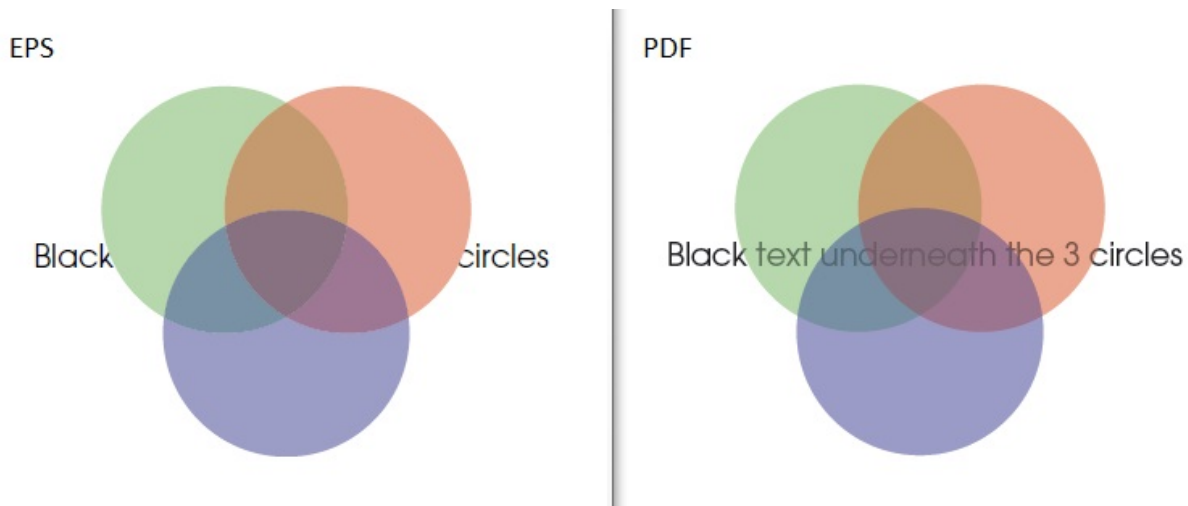
- Printing the VIPP application on a legacy PostScript RIP
- Printing the VIPP application on an APPE RIP
- Producing a PDF from a VIPP application (VI eCompose or PDF Export)

#### PRINTING THE VIPP® APPLICATION ON A LEGACY POSTSCRIPT RIP

VIPP® can use TIFF, JPEG, EPS, and PS files as image resources for VIPP® applications targeting a PostScript RIP. The exception to this is when using VI Design Express, where any Adobe InDesign image format including PDF can

be used for static (not for variable) images. This is because, when a VIPP® application is exported using VI Design Express, all static images are converted to EPS image files.

Prior to the VI Suite 11.0 release, PDF resources could not be used as variable resources in a VIPP® application targeting a PostScript RIP. With VI Suite 11.0 or above software, PDF files can be used as static or variable resources in VIPP® applications targeting the PostScript RIP by following the process to embed EPS information into the PDF file. Once a PDF resource file has gone through the process to embed EPS image information into the PDF, VIPP® will be able to process that PDF image in the PostScript RIP by using the EPS embedded information. However, when using this process, advanced imaging features such as transparency will be preserved inside the EPS image itself (flattening) but the transparency attribute will be lost. Thus, when printing to the PostScript RIP, transparency interactions with other parts of the page, if any, will be lost (to keep transparency, etc., you should target the APPE RIP).



The process to embed EPS information into PDF resources will increase the size of the PDF file used in the application. Once EPS information has been embedded into the PDF, this process does not have to happen again. However, there is no option to remove the EPS information from the PDF resource file.

### Embed EPS Information

The process to embed EPS information into the PDF resource(s) is enabled using either VI Design Pro or VI Design Express. This release does not include a standalone process for embedding EPS information.

#### Embed EPS information with VI Design Pro

VI Design Pro (VDP) is an Interactive Design Environment for generating VIPP® applications. Customers using VI Design Pro need to be competent in the VIPP® programming language. VI Design Pro provides a menu option under the File menu, Modify PDF with Embedded EPS, to embed EPS information into the PDF resources. This option will allow you to browse to a folder containing the PDF files you want to embed EPS information into. You may select one or more PDF files. Select Open to start the process. This could take several seconds per PDF selected. The PDF with embedded EPS information will increase in file size.

If you select not to embed EPS information using this option, they will be embedded automatically under the hood in temporary copies of the PDF files (in order to properly display the PDF image in the VDP window) and you will be prompted when you save your application to replace the original PDF files.

#### Embed EPS information with VI Design Express

VI Design Express (VDE) provides two options to process PDF resources and add EPS image information into a PDF.

- **During the Export of a document**

During the Export of a VI Design Express application, the export process will identify PDF resources that have not been modified with EPS information and add EPS information to each PDF. This process can take a few seconds for each PDF depending on the size of the original PDF image. The resulting PDF will increase in size because of the added EPS information. The time it takes to embed EPS information into each PDF will also add time to the Export function.

- **During a Batch process (recommended) prior to the design of a document**

The batch process is recommended as this is easier to manage and will overall take less time than the Export option. To batch process PDF resources, copy the resources to a folder. Access the PDF Batch Processing option from the VDE Options menu. Then browse to the folder location containing the PDF files. When the folder is selected, the batch process automatically begins. Depending on the number of PDF files in the folder, this process can take several seconds or minutes. The status of the files being processed is indicated on a status bar. Once the PDF files have been processed in this manner the PDF files do not have to go through this process again and the Export function will not flag these files as needing EPS information. Once this process has been completed, move the PDF resources to your asset folder and use the PDF resource as you would any other image.

VI Design Express will render the PDF file in the Adobe InDesign GUI, but will use the EPS information when printing to a VIPP®-enabled PostScript RIP.

## PRINTING THE VIPP® APPLICATION TO A POSTSCRIPT PRINTER

Printing a VIPP® application using PDF resources to a PostScript printer with VI Compose 11.0 and above will generate printed output. VI Compose will identify that it is running in a PostScript RIP and will read the EPS information in the referenced PDF file. It will use that information to render the PDF image. If the original PDF file had no advanced PDF features such as transparency, you will notice no difference in the output. If the original PDF used advanced features, because you are printing the EPS information on a PostScript RIP, those features, such as transparency interactions with other parts of the page, will be lost.

If your job uses PDF resources that for some reason do not have EPS information embedded into the PDF file a gray text box with the image file name will print instead of the PDF image. This indicates that you need to supply either a PDF resource that has embedded EPS information or call an equivalent version of the image in a supported format. Failing to do this the job will abort after 10 pages to avoid wasted paper on production jobs.

When printing to a PostScript RIP, it is recommended that you do not use PDF resource files with advanced PDF features. Such features are only supported by the APPE RIP.

## PRINTING THE VIPP® APPLICATION ON AN APPE RIP

The APPE RIP can process PDF resource files natively so no embedding of EPS information into the PDF resource is required when targeting the APPE RIP.

Because VI eCompose and APPE support PDF resources, PDF resource files with advanced PDF features such as transparency are supported.

PDF/VT is the variable implementation of PDF for the APPE engine. VI Compose will insert media and finishing information into a PDF using PDF/VT. The VIPP® application, when printed to the FFPS APPE, will pass media and finishing requirements to the FFPS.

VIPP® applications processed by VIeC will also generate PDF files with embedded media and finishing options. PDF

files generated when using the "Export to PDF" options in VDE or VDP will also have embedded media and finishing options in the PDF file. However, the media and finishing commands will only be supported when printing to an FFPS with APPE.

When designing a VIPP® job using VI Design Pro or VI Design Express, it is likely the tool will ask if you want to embed EPS information into the PDF. Only say yes if the application, at some point is printed to a PostScript engine. If you know that the job will only be printed to VI eCompose or FFPS APPE then it is useless to embed EPS information into the PDF resource. A VIPP® job targeted for the APPE RIP can use PDF, EPS, JPG, TIF and PS image resource types.

## PRINTING THE VIPP® APPLICATION TO FFPS APPE

Printing a VIPP® application using PDF resources to FFPS APPE will generate printed output.

The FFPS APPE RIP with PDF/VT support requires VI Compose 12.0 (or above) and FFPS 93 (latest available version or above). Failure to use these software versions will result in printing problems. VIPP® has been verified printing to the FFPS APPE engine, and EFI or other APPE printers are not currently supported.

A VIPP® job with PDF resources that you intend to process through the FFPS APPE print path requires some additional configuration. Either a VIPP® marker must be present in the VIPP® code, or, the FFPS Queue settings must be configured.

### VIPP® Code Marker

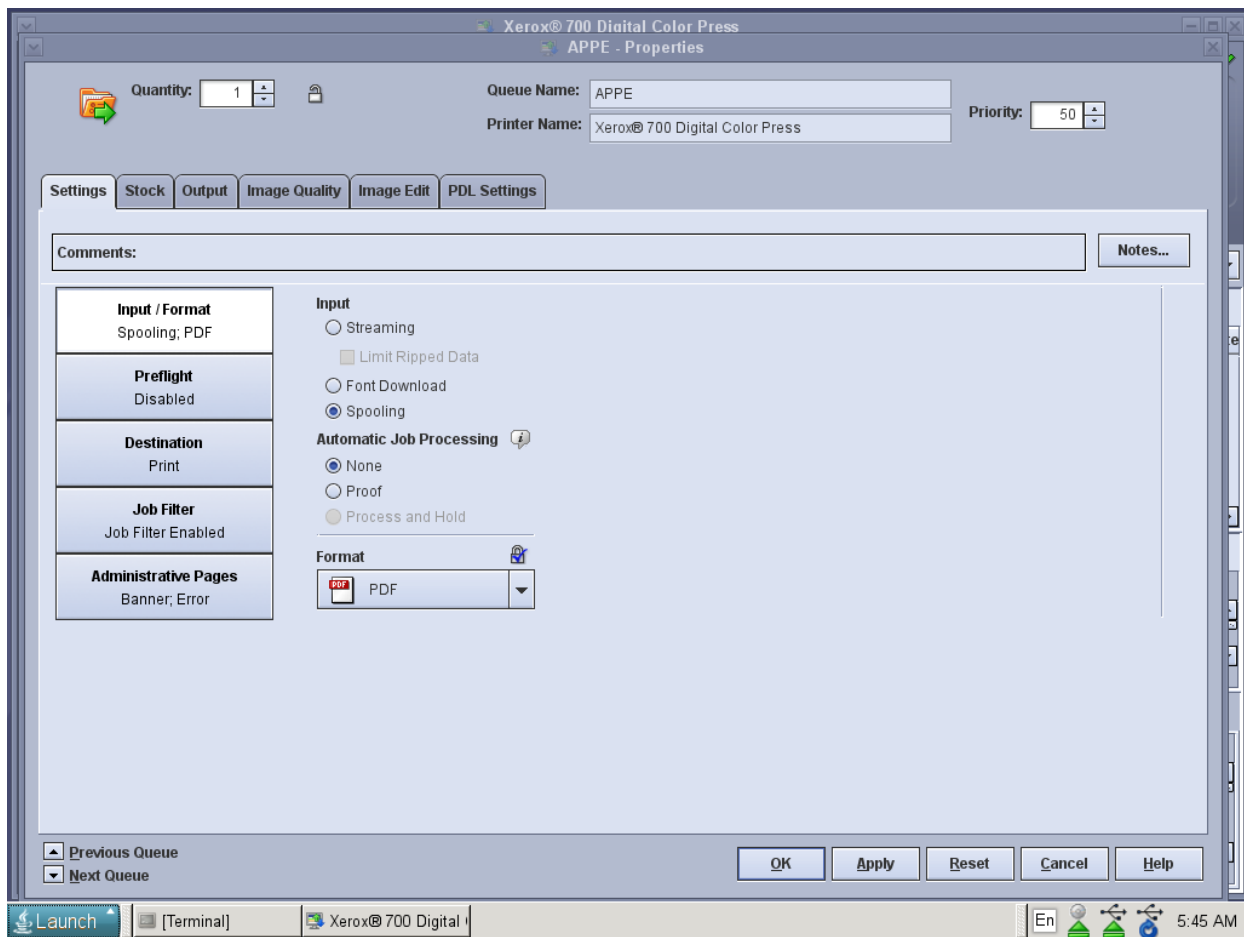
A VIPP® marker (% UsePDFXObjects: true) can be added to the submission file manually using VI Design Pro. If the VIPP® application is generated using VI Design Express and PDF resources are detected in the job, the tool will add the marker automatically to the VIPP® code. When present, FFPS will act on this marker and route the VIPP® application to the APPE print path.

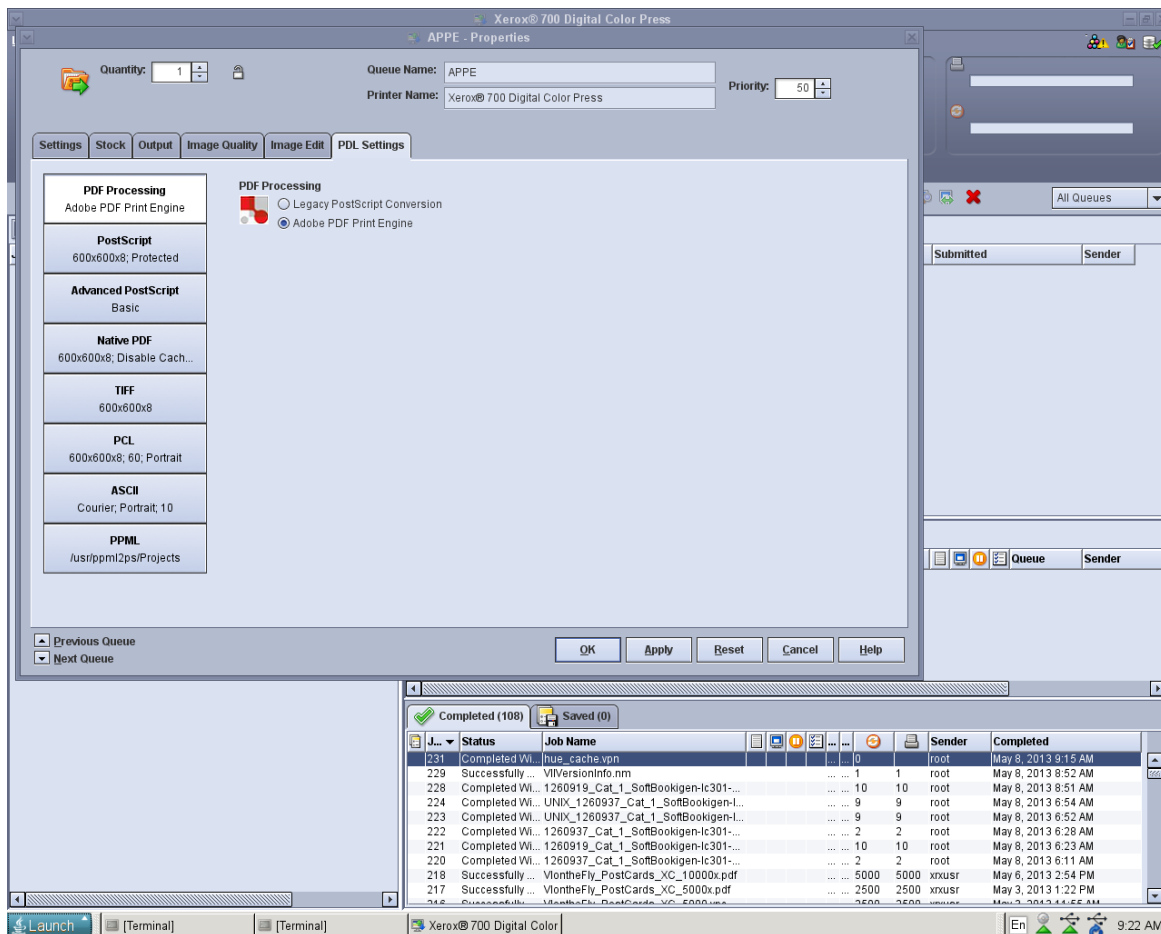
### FFPS Queue Settings (Recommended)

This is the recommended method for routing VIPP® jobs with PDF resources to the APPE print path. Two queue options need to be set:

- The Format option should be set to PDF and the override option checked
- The PDF Processing option should be set to Adobe PDF Print Engine.

VIPP® jobs without PDF resources can also be sent to this queue if required. For more information on these settings check with your FFPS analyst.





VIPP® jobs with or without PDF resources that are processed through the APPE print path will take slightly longer to process and appear in the FFPS processing queue.

## PRODUCING A PDF FROM A VIPP® APPLICATION, VI ECOMPOSE, OR PDF EXPORT

Processing a VIPP® application using PDF resources to a VI eCompose server or PDF Export functions (VI Suite 12.0 and above) will generate PDF output. VI eCompose will identify that it is running in an Adobe Normalizer interpreter and will use the PDF information instead of the embedded EPS information. This is because Adobe Normalizer can add references to external PDF resources in the PDF output. If the PDF resources contain advanced PDF features such as transparency these will be honored. The default action for VI eCompose is to embed all externally referenced PDF files in the final PDF file generated when processing a VIPP® application. You can change this behavior on a job by job basis or as a global change with the **SETPARAMS** command. For more details review the **SETPARAMS** command in the *VIPP® Language Reference Manual* and look for the parameter option **/PDFXembed**. It is recommended that you do not change this default.

## ADOBE ACROBAT AND ADOBE READER INFORMATION

A PDF output created out of a VIPP® application using PDF resources, often referred to as PDF referenced XObjects, can be viewed using Adobe Acrobat or Adobe Reader under the following conditions:

1. The version of Adobe Acrobat or Adobe Reader must be 10.1.6 or above as 10.0 does not work.
2. You must open the PDF output from an Adobe Acrobat/Reader privileged folder. To declare such a folder you must edit the following Acrobat/Reader options in the Preferences menu. See screen shots below:

- Category: Page Display

Under Reference XObjects View Mode set **Show reference XObject targets to Always**.

and

- Category: Security (Enhanced)

Under Privileged Locations click on **Add Folder Path** and set Location of referenced files to the directory containing the PDFs generated by VIeC or any parent directory.

If, when viewing the PDF output, you see gray text boxes with the image name instead of the rendered image, then one or more of the following conditions is true:

- the version of Adobe Acrobat or Reader does not support references to XObjects
- the settings in Acrobat/Reader as discussed above are not correct
- the EPS information was not embedded in the PDF resources
- the referenced PDF is not accessible. External reference is not valid if you choose not to embed NOTES

#### Select the VIeC incoming folder

Under Location of referenced files, it is recommended that you select the VIeC incoming folder instead of the folder containing a specific job. As long as all of the PDF's generated by VIeC are under the same incoming folder hierarchy then all referenced PDFs should be found and properly displayed.

#### PDF referenced XObjects are independent of the Enhanced Security checkbox.

The state of the checkbox does not govern viewing of PDF referenced XObjects

### PRINTING PDF FILES WITH REFERENCED XOBJECTS

A PDF output with referenced XObjects produced by a VIPP® job is primarily intended to be printed on an FFPS APPE. If such a PDF is printed on a PostScript printer it will first undergo a PDF to PostScript conversion. In most cases this conversion will ignore the PDF references and image the XObject using the EPS information (if present) or print a gray text box.

## VIPP® and Variable Information Suite Applications

VIPP® functionality provides the foundation on which these Variable Information Suite applications are based:

All of the Variable Information Suite applications are described in more detail in their respective User Guides.

### **VI Design Pro (VDP)**

Provides an Interactive Design Environment (IDE) for VIPP® application design. A limited GUI interface and wizards are available to assist in the creation of VIPP® applications. A Smart Editor provides command syntax assistance. The three panels of VI Design Pro allow you to see the data driving the application design (line, delimited, or XML data), a GUI representation of the application in real time, and the VIPP® Code. Changes made in the GUI will update the VIPP® code. Changes to the VIPP® code will update the GUI display.

VI Design Pro requires knowledge of the VIPP® programming language. If you are not a programmer, ask a Xerox representative about third-party GUI design tools that can be used to create VIPP® jobs. Most of these tools provide VIPP® design capabilities using a drag and drop interface and require no prior VIPP® programming experience.

### **VI eCompose (VieC)**

Provides the ability to create files in Adobe PDF format, complete with bookmarks and links, then dispatch those files to any defined location (directory folder, email, print device, etc.) via a user defined process. Administer the VIPP® jobs and VI eCompose Web Servers remotely from any PC with permission to access those servers. Provides a Web Job Submission Service (WJSS) to create large numbers of small jobs in Adobe PDF format.

### **VI Design Express (VDE)**

A plug-in application for InDesign Creative Suite (Mac OS/X and Windows), that allows designers with minimal programming experience to quickly create VIPP® jobs using delimited or XML data.

### **VI Explorer (VIE)**

Provides a GUI interface, which allows programmers to preview and verify VI jobs and projects, and to provide easy access to Normalization and Demographics services.

### **VI Compose Services**

Provides functionality in addition to the normal workflow.

VI Compose services provide demographic information about the jobs for reports and tracking, and a normalization feature, which transforms VIPP® jobs so that they can be pre-processed before they are printed or imaged.

## **VI COMPOSE FUNCTIONALITY**

VI Compose is device-resident and requires no extra virtual memory. VIPP® resources, such as forms and images, can be retained on the printer or network. Network traffic and production time are reduced because only the unformatted variable data file with the initial start commands is required for printing and imaging.

In VI Compose, VIPP® commands are a set of page layout functions bundled as a PostScript prolog, which allow imaging and data processing applications to produce a data stream output supported by any VIPP®-enabled PostScript output device.

VIPP® data streams must be processed by a PostScript interpreter, which accesses resources such as VI Compose, forms, images, and JDTs, and other required resources.

VI Compose processes data streams in these modes, which are described in detail in [VIPP data streams](#):

- Native mode
- DataBase mode
- Line mode
- XML mode

The VIPP® Language allows you to enhance, reformat, and colorize legacy output (line data) from COBOL or CICS programs, generated by the data center and targeted toward fan-fold line printers. In addition, the VIPP® Language has composition features that handle other data streams. Delimited database data, tagged or fixed record data, even-tagged delimited data, such as output from SAP or other client server type applications, can be formatted using the VIPP® Language.

### **VIPP® and VI Compose features and functions**

VIPP® and VI Compose features and functions:

- Provide optimal throughput and efficiency for image and data processing applications.
- Support resource pre-rasterization and caching.
- Allows use of PDF, EPS, TIFF, JPEG, PostScript files, and text files (TXT) files as variable resources. Refer to [Using PDF resources with VIPP \(APPE, PDF/VT, and Embed EPS options\)](#) for details on PDF use with VIPP®.
- Provide direct access to PostScript PDL, which supports rich TIFF images, curved boxes, and PostScript fonts with scaling and rotation.
- Speed the page composition process by composing the pages on the imaging device when the imaging occurs. Since this is not a conversion process, no intermediate files are produced.
- Allow VIPP® jobs to remain independent of the evolution of PostScript. VI Compose can use higher- level PostScript operators as soon as they become available allowing you to take advantage of performance improvements without modifying the jobs.
- Allow printing without converting documents.
- Allow you to print line mode jobs on PostScript-enabled devices using PostScript functions without altering print files.
- Enable PostScript formatting and printing of database and XML files, including forms and logos, on supported PostScript devices.
- Convert character-based information to Code 39, Code128, EAN, UPCA, 2OF5, POSTNET, PDF417, MAXICODE, DATAMATRIX, Aztec, QRcode, and IMB (4-State barcode) barcode output.
- Save network bandwidth since the formatting files and resources (for example, images) can be stored at each printer and only the variable data portion of the document is sent for each print run.
- Allow you to take full advantage of powerful PostScript features without investing in costly PostScript training.
- Allow production of data-driven graphics from variable input data.
- Provide text highlighting using lines, boxes, shading, and outlines.
- Enable text reflow across frames, including pages.

- Print multi-page PDF, TIFF, Decomp, or FreeFlow Makeready RDO files, allowing the assembly of pre-composed document segments into a larger document using conditional logic and data values to drive construction of the document.
- Allow you to combine numeric variables, numeric constants, and arithmetic operators into a single operand using arithmetic expressions.
- Provide expanded support for date and time using Commands and variables.
- Include solid coated and uncoated color simulation as supported by FreeFlow Print Server using VIPP® list of predefined Colorkeys.
- Support the application of a color tint to any CMYK or RGB Colorkey argument to a VIPP® command referencing a Colorkey.
- Support conditional text, forms, images, and pages.
- Provide font attributes and numeric format commands.
- Provide ZSORT capabilities (Imposition).
- Provide date and time functions.
- Support horizontal and vertical printing with multi-byte fonts (primarily Chinese, Japanese and Korean (CJK) fonts).
- Support for Specialty Imaging.
- Support for printing Arabic text strings.
- Cutmarks - places marks on the paper for trim operations
- OMR - supports both bubble forms and automated mail OMR
- 4-State barcode - support for the 4-state barcode

In addition to the features and functions of VI Compose, VIPP® functionality has been expanded with the addition of the following programs:

- FreeFlow® VI Design Pro (VDP)
- FreeFlow® VI eCompose (VIeC)
- FreeFlow® VI Design Express (VDE)
- FreeFlow® VI Explorer (VIE)
- FreeFlow® VIPPManage (VM)

## **FREEFLOW® VI DESIGN PRO**

VI Design Pro includes a Graphical User Interface (GUI), application templates, and Smart Editor features that simplify the creation and modification of the VIPP® jobs or VI Projects. VDP provides full support of the VIPP® command set in order to create new VIPP® jobs or VI Projects or to view and modify existing VIPP® jobs or VI Projects created in one of the following modes:

- Native mode
- Line mode
- Database mode

- XML mode

VI Design Pro provides a What You See Is What You Get (WYSIWYG) representation of the job. This provides an interactive environment in which to create the VIPP® jobs or VI Projects, and represents how the job will print on the output device.

VI Design Pro is the only GUI-based product that can use all of the capabilities provided by VIPP® to process data-driven graphics, data-driven conditional processing, transactional printing, and database publishing. However, there are third-party GUI design tools that can be used by non-programmers, which provide VIPP® design capabilities using a drag and drop interface.

### VDP features and functions

The features and functionality available from VI Design Pro provide the following benefits:

- An easy-to-use GUI to help you access, view, modify, and create VIPP® jobs or VI Projects
- Full support of the VIPP® command set using GUI capabilities (with the exception of demographics and normalization, which are not intended for use with VI Design Pro)
- Availability of powerful PostScript features without costly PostScript training
- Templates that assist in creating new jobs
- A VI Project feature to bundle project components into physical and logical groups
- An interactive Smart Editor feature to easily add or modify VIPP® commands or edit command parameters and values in source files
- Easy access and viewing capability for all files and resources associated with the job
- Maximum control of the document layout
- Immediate WYSIWYG display of output changes without having to print the job
- Automatic generation and display of VIPP® code as changes are made
- Terminal Services provides the capability to access VDP via Remote Desk Top
- Can generate PDF output directly from the design tool, or can generate .vpc or VIPP® templates for fast printing on VIPP®-enabled devices

### FREEFLOW® VI ECOMPOSE

VI eCompose allows you to generate documents in Adobe Portable Document Format (PDF) from VIPP®-based variable data jobs. The *FreeFlow VI eCompose User Guide* provides an overview of the features and functions available, describes different installation configurations, and explains how to use the software.



Note: Throughout the Variable Information Suite Documentation, the term PDF refers to Adobe Portable Document Format files; Distiller is an Adobe Acrobat program.

VI eCompose processes VIPP® jobs to produce PDF output files. Use the following options to submit a VIPP® job to VI eCompose:

- VIeC Job Submission Client (including a web-based client)
- Watched Folders Client
- lpr

When you use the VIPP® **BOOKMARK** command in the VIPP® job, VIeC splits the PDF output file into smaller PDF

files at bookmark boundaries. This can be used to create fully customized PDF sets per customer record.

In addition to using bookmark contents as page split delimiters, use bookmarks to determine the format of the PDF output file names and include them in the correlating index file produced with the job.

Thus, there are several possible workflows:

- In the absence of bookmarks, use VIeC to create the PDF file(s).
- Distill with bookmarks to create the PDF file with bookmarks and defer the split (making it a separate step). Optionally, split the PDF file with bookmarks into smaller PDF files with an index file.
- Distill with bookmarks and split as one integrated process, to create the PDF file with bookmarks and then split that file into smaller PDF files with an index file.

As each job is processed, billing meters on the appropriate VIeC Server are incremented to track the number of pages distilled, pages split, and PDF files generated. A server-level accounting file also tracks resources used on a per-job basis.

VI eCompose functionality can be accessed either directly at a VIeC Server, or remotely using VI eCompose Web and an internet browser. With VIeC Web, VIPP® jobs can be submitted to a web-enabled VIeC Server, create new or additional VIeC user accounts, and administer VIeC Servers via a web browser.

VI eCompose output can be dispatched to other programs or functions such as email, fax programs, or a DocuShare repository using VIeC Dispatch. Dispatch monitors VIeC output and performs post-processing via a customer-specified back-end. Post-processing is completed in accordance with parameters and other data embedded in a job, and is communicated to the dispatch mechanism via the index file of completed VIeC jobs.

### VIeC features and functions

VIeC seamlessly integrates the capabilities of VI Compose with the Adobe Acrobat. It provides numerous enhancements over previous VIPP® to PDF solutions. VIeC:

- Accepts as input not only standard VIPP® print files (line mode, database mode, native mode, XML mode), but also VIPP® jobs submitted as VI Project Containers (.vpc).
- Splits output into multiple PDF files at specified page boundaries through the use of the VIPP® **BOOKMARK** command.
- Generates a per-job index file to assist you in correlating specific output files with their corresponding input records.
- Generates an accounting file to enable you to track jobs submitted, page counts per job, and other job information.
- Tracks the number of pages distilled, pages split, and output files generated via three separate billing meters.
- Enables clients to submit jobs for processing both locally and remotely using the CPU/disk of a VIeC Server on a network.
- Enables you to configure multiple VIeC Servers to cooperatively process jobs submitted to a common file Server (in this configuration, the combination of the shared file Server and VIeC Servers forms a VIeC Cluster).
- Allows you to use VIPP® PDF Interactive Features, a set of VIPP® commands that allows you to create interactive elements when the VIPP® job is rendered into a PDF document.
- Allows you to submit jobs to VI eCompose via lpr, enabling you to use VIeC from a mainframe host and/or enabling automated workflow.

- Allows you to submit jobs to VI eCompose from user-defined watched folders, and to ensure that the submitted files are ready for processing using the file ready check option.
- Allows you to direct the PDF output of multiple jobs to a single bulk directory to facilitate batch or other automated post-processing of VIeC generated output.
- PDF Splitter module supports PDF Security. Some PDF-Security permissions are passed from the main PDF document to all child PDF files generated during the split process. This includes unique PDF passwords per child PDF.
- When a job submission is successfully completed, the PDF documents generated can be found and viewed by selecting the job folder entry in the Job Status list and double-clicking to invoke a File Selection dialog.
- VIeC Dispatch provides a mechanism by which VIeC jobs can be sent to other software such as DocuShare, email, and fax programs.
- VIeC Web allows you to submit VIeC jobs to remote servers without the need for specialized job submission client software other than a web browser, resulting in less software for network administrators to maintain.
- VIeC Web and HTTPs can be used to submit VI eCompose jobs through network fire walls where mapping or mounting a remote drive might not be allowed, making remote VI eCompose Servers available to users with the appropriate access permissions.
- Provides Multiple Instance support for VIeC and or VIeC Dispatch engine on a single server.
- Cloud ready Web Job Submission Server (WJSS) version available in the standard configuration, for more information, refer to the *FreeFlow VI eCompose User Guide*.

## FREEFLOW® VI DESIGN EXPRESS

VI Design Express (VDE) is an application plug-in for use with Adobe® InDesign and Adobe InDesign Creative Cloud. For information on the supported versions, refer to the *FreeFlow VI Design Express User Guide*.

Available as a download, VDE is designed to allow a graphic artist to:

- Create applications that contain variable information.
- Export those applications to a VI Project Container (VPC) file.
- Print the application at production speed on a Xerox VIPP®-enabled print device.
- Process the VPC using VI Design Pro, VI Explorer, or VI eCompose.
- Emit PDF files directly from VDE.

In addition, VDE allows the graphic artist to do all of that without having to learn VIPP® programming techniques or the VIPP® language.

VDE is available on both Mac OS X and Windows platforms, and is designed to facilitate creation of variable data applications that take advantage of the speed the VIPP® software provides, from within a familiar design environment.

## FREEFLOW® VI EXPLORER

VI Explorer is a Graphical User Interface (GUI) program designed to allow programmers to preview and verify VI jobs and projects, and to provide easy access to Normalization and Demographics services.

VI Explorer:

- Allows viewing and verification of VIPP® jobs.
- Allows printer operators to identify what resources a job might require via Demographics.
- Allows printer operators to break the job up into pieces by generating a page-independent version of the job via Normalization.
- Integrates Demographics and/or Normalization into a workflow by using the Watched Folders support for Normalization/Demographics that is shipped with VI Explorer.
- Terminal Services provides the capability to access VIE via Remote Desk Top

#### **FREEFLOW® VIPP® MANAGE**

VIPP® Manage is a Windows based utility that provides a GUI for the management of VIPP® resources on the disk of a VIPP®-enabled desktop or office print device, and for managing VIPP® jobs and resources on diskless printers.

## FreeFlow® VI Compose Open Edition

The VIC(OE) software is a modified version of VI Compose software supported on Xerox print engines. The VIC(OE) software has been modified to install on non-Xerox Windows devices and to support license activation through normal Xerox channels.

Due to internal operational differences between Xerox and non-Xerox Windows print devices, some limitations will apply. These limitations exist mainly around the areas of feeding and finishing, but can also exist in other areas. For example, in jobs where VIPP® will attempt to write file position information to the device. Because Xerox has no control over the third party devices, operations that are normal for Xerox devices may not be allowed on the third party device. Because of these possible limitations, Xerox recommends that all jobs run to a third party device are fully validated before running in production.

Because Xerox has no engineering support for non-Xerox Windows production devices, any issue reported will be validated against a similar Xerox production device. If the Xerox production device exhibits the same issue, Xerox will fix the issue in a software patch and provide the fix to the customer to install and verify. If this does not fix the issue on the non-Xerox Windows production device, Xerox may be unable to assist further.

For more information, refer to *FreeFlow® VI Compose (Open Edition) Installation and Overview*.

# VIPP® Data Streams

This chapter contains:

Native Mode .....	75
Line Mode .....	76
Database Mode .....	78
XML Mode .....	80
Prefixed Delimited Transactional Data .....	83
Linking the Data with the JDT, DBM, or XJT .....	84
Record Structures .....	85
Record Structures and Mode/Presentation Relationship .....	87
Functions .....	88
Conditional Processing .....	90
Dynamic Boxes .....	91
Data Driven Graphics .....	92
Native Mode Prefix .....	93
Distribution List .....	94
End of File .....	95
Coding Examples .....	96

VIPP® jobs can be sent to print in one or more of these modes:

Native mode	The default, a PostScript data stream comprised of VIPP® commands only.
Line mode	A non-PostScript data stream processed by the single VIPP® command STARTLM. Line mode is page oriented; it reads a page of data at a time and prints that data using either listing or RPE presentation options as defined in a Job Descriptor Ticket (JDT).
Database mode	A non-PostScript data stream processed by the single VIPP® command STARTDBM. Database mode is record oriented; it reads one record at a time and calls the Data Base Master (DBM), which processes the record data. Each call to the DBM can use the record data to print one or more pages, one or more lines, or store the record data in memory for use at a later time.
XML mode	An XML data stream processed by the single VIPP® command STARTXML. XML data is parsed and then printed on pages according to the layout instructions contained in an XML Job Ticket (XJT).



Note:

### Prefixed Delimited Transactional Data

While not a VIPP® mode, it is listed here to describe a capability in the VI Design Express design tool plug-in to Adobe InDesign to generate simple to moderate complex transactional documents driven by prefixed delimited transactional data. VI Design Express can also be used to create promotional documents using delimited or promotional XML data.

The purpose of line mode, database mode, and XML mode as opposed to native mode, is to keep data production and data presentation as independent as possible, allowing you to design the layout and update the jobs without modifying the application.

Line mode, database mode, and XML mode also enable processing of data including conditional processing at the page, record, and field/tag levels at the target VIPP®-enabled device. This allows you to send data files directly to the printer, bypassing lengthy composition steps on the host. The primary advantages of this method are:

- Faster production of the data file at the host
- Minimum amount of data sent over the network
- Optimal processing speed of PostScript at the printer

The data in line or database mode jobs is further organized into records, or lines, which may have one of these structure variants:

- Variable or fixed record length
- Control codes or Printer Control Characters (PCC)
- Line printer or field delimited data
- Non-prefixed or prefixed data

All four variants may be combined, which provides 16 different record structures.

VIPP® native mode, line mode, database, and XML mode commands are described in the *VIPP® Language Reference Manual*. Many VIPP® commands are available in all three modes.

## Native Mode

VIPP® native mode refers to files comprised of VIPP® commands only. These commands are used to place text, images, graphics, logos, forms, and segments directly on a page. Native mode can be used for these purposes:

- To produce a VIPP® native mode data stream that directly controls the page composition process. Production of the data stream is accomplished using an application, a converter, or a postprocessor.
- To encode resources such as forms, segments, JDTs, and DBMs. Resources are encoded by the layout designer using a text editor or any advanced VIPP® user interface such as VI Design Pro.

Native mode commands may also be embedded in line mode data streams. Refer to [Native mode prefix](#).

VIPP® native mode files are processed directly by the PostScript interpreter. There is no explicit native mode invocation for these files. For this reason, native mode is the default VIPP® mode.

### TIP

To avoid confusion, use the .nm extension for native mode jobs, and the .lm extension for line mode jobs.

## Line Mode

Line mode refers to legacy type data streams, such as LCDS jobs, and is sometimes referred to as line or print ready data. For more information, refer to [LCDS migration](#). Data typically consists of ASCII or EBCDIC with fixed or variable length records, and also includes prefixed (tagged) and/or field delimited data streams. Line mode allows you to read, process, and format the line printer data, line- by-line and column-by-column. It is most often used for transactional type jobs such as a telephone billing statement.

Use the VIPP® **STARTLM** command to start line mode.

With line mode, the page composition of the line printer data is controlled by a JDT, which is provided as a parameter to the VIPP® line mode initiator **STARTLM** command. A JDT is a file that contains the VIPP® commands used to define the desired presentation of the line printer data.

Line mode is further divided into two presentation options, defined in these sections:

- [Listing](#)
- [Record processing entry](#)

### LISTING

The listing presentation option is the line printer emulation presentation used to print internal sysouts that do not require sophisticated presentation capabilities.

When this option is used, page layout features such as orientation, margins, grid, font, and form are defined globally for a page using a JDT. In addition, page numbering, automatic zebra striping, and frame generation can also be used.

The automatic font scaling feature allows scaling of a fixed pitch font according to the grid, which is the number of Characters Per Line (CPL) and number of Lines Per Page (LPP). For more information, refer to **SETFONT**.

Page delimiting is controlled by setting a maximum number of LPP or using an explicit page delimiter such as Form Feed (FF) or Skip to channel one when PCC is used. For more information, refer to **SETGRID**, **SETPCC**, **SETPBRK**, and **SETSKIP**.

The listing presentation option is initiated by a **STARTLM** sequence invoking a JDT with listing specifications.

### RECORD PROCESSING ENTRY

The Record Processing Entry (RPE) presentation option allows each record to be split into fields that can be printed any number of times and at any location on the page. RPEs can contain specific presentation attributes such as position on the page, line spacing, font, color, alignment, and rotation. The RPE presentation option is initiated by a **STARTLM** sequence, which invokes a JDT that contains RPE definitions.

The RPE fields are defined horizontally using one of these:

- character position and field length
- field number

The RPE fields are defined vertically using one of these:

- line number (FROMMLINE)
- prefix (SETRPEPREFIX, RPEKEY)

Refer to FROMLINE and RPEKEY for more information.

Field definitions, location on the page, and presentation attributes for a given group of lines or a given prefix are coded in an RPE definition. Page delimiting is controlled the same way as in the listing presentation option. Line printer files built with one data record per page, referred to as unformatted records, can also be processed, one line per page, using this option.

## Database Mode

Database mode is used to process a single file containing delimited records. The delimited file is typically created as an extraction or query from a relational database. Jobs processed using database mode include mail-merge or promotional jobs.

Database mode uses these file types:

- Database file
- Data Base Master (DBM) for processing variable fields
- JDT for global page layout definitions (optional)

Records in a database file may be fixed or variable in length, and must have the same sequence and number of fields. Use the SETDBSEP command to define the field delimiter as any character such as a comma. The structure of the database file is defined in the first record, which consists of the field names and their sequence within the record.

A Data Base Master is a template file in which field names are used as place holders for each variable item. It can also reference other VIPP® resources such as segments and images. The DBM is coded in VIPP® native mode. PostScript cannot be used to code a DBM because PostScript drivers do not allow place holders. PostScript or EPS files can be called as forms or segments in a DBM.



Note: Use .dbf and .dbm extensions for database files and Data Base Masters to facilitate file type identification.

Database mode is initiated by submitting a database file beginning with a STARTDBM sequence. STARTDBM passes a DBM name to VI Compose. VI Compose reads the first record of the database file and registers the field names. For subsequent records, VI Compose replaces the DBM field names with the appropriate value, then recomposes and images each page of the DBM. Dynamic composition of the page occurs when lines and paragraphs using variables from the database file are reformatted. Variables can also be used to reference forms, logos, and media.

When no DBM name is provided to the **STARTDBM** command, a DBM name must be supplied with each record in a DBM\_NAME field. Use this capability to process a different DBM for each record.

Page transition is controlled by the **PAGEBRK** command in the DBM. Page overflow is controlled by the SETLKF command.

Reference database fields in the DBM using one of these methods:

- Use a field name to directly replace a string operand of any native mode command such as **SHx**, **SETFORM**, **SETMEDIA**, **ICALL**, or **SCALL**.
- Embed the field name(s) in a string, encapsulated between "\$\$" and ".", or between "[=" and "=]", and followed by the VSUB command. VSUB replaces each field name by its value when the DBM is processed. Use this method when a field name must be merged with other fixed text, like a field name occurring in the middle of a paragraph.

Global page layout definitions (such as duplex, page size, Multi-Up, index definitions, and others) can be executed once along with the DBM. Place the definitions before the **STARTDBM** command either in the database file or grouped in a JDT called by SETJDT, or include them in the DBM using an IF/ENDIF construct.

## SELECTING DATABASE FIELD NAMES

A database field name must be recognized by VI Compose as being a valid (legal) field name. All field names must be defined using 7-bit ASCII characters and must not contain spaces or any of these characters:

% / \ ( ) { } [ ] < >

Failure to use a valid field name can result in VI Compose errors or unexpected results. To avoid these issues when defining a field name, Xerox recommends use of the following syntax:

XXXXzzzzzz

**Where:**

- X** is one or more uppercase alphanumeric (ASCII) characters.
- z** is one or more lowercase alphanumeric (ASCII) characters.

Avoid using database names that contain all uppercase or all lowercase characters as doing so may inadvertently create a field name that conflicts with a PostScript or VIPP® reserved word. This conflict creates an illegal field name. Also avoid field names that use all numeric characters or strings that could be interpreted as a numeric value, again these are considered illegal field names.

**Examples of valid field names:**

- FName
- FIRSTname
- Form101

**Examples of field names to avoid:**

<b>fname</b>	all lower case field names can conflict with PostScript Reserved Words
<b>FNAME</b>	all upper case field names can conflict with VIPP® Reserved Words
<b>(Fname)</b>	field names cannot be enclosed between ( ), { }, [ ] or < > characters
<b>F name</b>	field names cannot contain spaces
<b>101</b>	will be interpreted as a numeric value and is an invalid field name
<b>16#10</b>	is treated as a radix number and is an invalid field name
<b>123E5</b>	is treated as a real number and is an invalid field name
<b>Any non ASCII string</b>	field names must be comprised of ASCII characters only

## XML Mode

XML mode enables the processing of XML data.

XML mode is initiated by a STARTXML sequence. This sequence can be placed at the beginning of an XML file together with an initial %! line or used in a submission file after a SETLMFILE sequence that references an XML file.

The name of an XJT must be provided as an operand to STARTXML.

An XJT is a file that contains instructions on how to process and arrange the XML data into a document. It is similar to a line mode JDT. It contains all global layout definitions for the document (orientation, forms, media, frames, fonts, colors, etc.) plus an XML Processing Definition (XPD) table (BEGINXPD/ENDXPD) that describes specific actions to be performed on specific XML tags (BTA/ETA, BTS/ETS).

### XML DATA STRUCTURE OVERVIEW

XML data is ASCII text data organized in a tree structure.

Each node of the tree has a name, or tag, encapsulated between left and right angle brackets (< and >). Tags are usually found in pairs consisting of a start-tag and an end-tag.

Each node may contain text between the start and end tags. Nodes with no contents may be compacted as a single tag called an empty-tag. The formats of the 3 types of tags are:

- start-tag <Tagname>
- end-tag </Tagname>
- empty-tag <Tagname/>

For more information on the XML language, refer to <http://www.w3.org/TR/REC-xml>

### XML VARIABLES

During the processing of an XML file VI Compose accumulates the node contents and places them into a dictionary of variables. Each variable is registered with a VIPP® XML Variable path (VXVpath). A VXVpath for a given node content consists of all the parent node names including the node itself preceded by a caret (^).

#### Example:

Assuming this XML data tree:

```
<invoices>
  <invoice>
    <customer>
      <fname>John</fname>
      <lname>Smith</lname>
      <phone type="home">111.111</phone>
      <phone type="work">222.222</phone>
    </customer>
  </invoice>
</invoices>
```

The node contents John will be registered with this VXVpath:

```
^invoices^invoice^customer^fname
```

When a node tag has attributes, the attribute values will be registered with a VXVpath created by concatenating the VXVpath of the node and the attribute name preceded by @. In the example above, the attribute values home and work will be registered with this VXVpath:

```
^invoices^invoice^customer^phone^@type
```

A VXVpath can be abbreviated to any sub-path extracted from its right part. For example, the first VXVpath above can be abbreviated to:

```
^invoice^customer^fname
```

Such a sub-path is called a VXVname.

A VXVname is mapped to its VXVpath according to the current context at the time it is used, refer to **BTA** and **BTS** commands. When a VXVname maps two or more VXVpaths, VIPP® chooses the closest match to the current VXVpath. When it is not unique the job will abort with the message:

```
VIPP®_ambiguous_name in _name
```

meaning that the VXVname must be additionally qualified to remove the ambiguity.

A VXVname can be used in the same manner as any variable name to replace a VIPP® command operand, or it can be embedded in a VSUB string.

When a VXVpath has not yet been created it returns an empty string. When a node is entered its VXVpath, and all previous VXVpaths that are children of this node, are reset. For these reasons VXVpaths do not need to be explicitly initialized or reset.

In addition to native VXVpaths created by the processing of the XML data, create custom VXVnames using the SETVAR or ADD / SUB commands.

A custom VXVname will be registered with a VXVpath created by expanding the VXVname with the current VXVpath. Thus, custom VXVnames benefit from the implicit initialization and reset of VXVpaths.

### Example:

Assuming the XML data above, the phone numbers may be registered by this code during the processing of the phone data:

```
BTA /customer^phone {}
{ CASE ^phone^@type {}
  (home) { /^customer^hphone ^phone SETVAR }
  (work) { /^customer^wphone ^phone SETVAR }
  ENDCASE
} ETA
```

In this example ^customer^hphone and ^customer^wphone are registered with these VXVpaths:

```
^invoices^invoice^customer^hphone
^invoices^invoice^customer^wphone
```

This more streamlined coding can be used to produce the VXV paths that follow:

```
BTA /customer^phone {}
{ (^customer^$$^phone^@type.phone) VSUB ^phone SETVAR } ETA

^invoices^invoice^customer^homephone
^invoices^invoice^customer^workphone
```

## PRINTING XML VARIABLES

The XML file may be encoded using any supported encoding, and the node contents, the text between the start and end tags assigned to XML variables can be printed with any **SHx** command. The font used must have been encoded (see SETENCODING) with an encoding matching the XML file.

XML files encoded using UTF-16 are supported, providing the XML data begins with a UTF-16 BOM (OxFFFE or OxFEFF). Note that UTF-16 XML data is converted to UTF-8 on the fly, so the node contents must be printed with UTF-8 encoded fonts.

## STRIPPING BLANK AND CONTROL CHARACTERS

Stripping duplicate and extra spaces and ignoring control characters such as CR, LF, and tab, is the regular behavior of an XML parser. There are two ways to change this behavior:

### The XML way

Place the data for which spaces and control characters must be preserved within CDATA sections encapsulated between special delimiters.

```
<![CDATA[ ...data ... ]]>
```

so for example:

```
<string3><![CDATA[ 3 spaces ]]></string3>
```

### The VIPP® way

Set the /TextFilter parameter to false to disable spaces and control character filtering. However, when this parameter is set to false, unexpected results may occur because the control characters used to structure the XML data (for view in a text editor for instance) are now part of the contents.

Use one of these examples to work around these effects:

- Add a third empty proc before ETA (partial contents action):

```
[/TextFilter false] SETPARAMS
BEGINXPD
BTA /root
{}
{ ^string3 SHL }
{}
ETA
ENDXPD
```

- Specifically set TextFilter for each node for which you want to turn off the filtering.

```
% turn filtering off for node 'string3' only
BTA /string3
{[/TextFilter false] SETPARAMS}
{[/TextFilter true] SETPARAMS}
ETA
```

## Prefixed Delimited Transactional Data

VI Design Express has the capability to create simple to moderately complex transactional documents driven by prefixed delimited transactional data files. For more complex transactional applications, VI Design Express can be used to create the base of the transactional document and the VI Design Pro tool or Xerox Professional Services can be engaged to add the complexity required.

Transactional documents deliver critical content to the end customer. Examples of transactional documents include Phone Bills, Credit Card Statements, Invoices, Inventory Reports, etc. Transactional documents that include cross selling advertising are called Trans Promo documents and can be used to add value to the transactional documents you create. Using VI Design Express and prefixed delimited data you can create these types of documents quickly and easily.

Below is an example of a prefixed transactional data file. This example shows three customer record sets.

```
H0,David,David L Kirk,36 Beech Road,Bushbury,Wolverhampton,West Midlands,ST10 9YF
H1,M2,5249 500 3658,0893 1272 7544 5087,04 March 2015,31MAR15,31 March 2015,30JAN -
28FEB,JAN15-FEB15,005352,-005000,000352
H2,005045,004999,001000,000670,001030,013096,00500,01500,00428,01267,01695,000126,000073,00013
3,000332,Evolvcom 700F,004667,,
T0,01JAN,9:00a,Humberside,2201 990 0003,24Hr,10,90
T0,02JAN,3:40p,West Aedale,0022 760 3303,24Hr,6,50
T1,01FEB,10:13a,Frankfurt,+49 621 41 12 15,24Hr,23,320
T1,02FEB,1:40p,Madrid,+34 515 14 13 12,24Hr,6,99
T1,03FEB,11:04a,Paris,+33 121 31 41 51,24Hr,10,150
H0,Helen,Helen E Tavener,23 Teasel Road,Fordhouses,Wolverhampton,West Midlands,ST10 8QA
H1,F1,7987 225 8232,7292 8840 2612 4303,04 March 2015,20MAR15,20 March 2015,19JAN -
20FEB,JAN15-FEB15,004357,-004357,000000
H2,004354,002999,000000,000478,000837,008668,00200,00900,00153,00724,00877,000126,000073,00013
3,000332,Evolvcom 400F,002667,,
T0,01JAN,9:00a,Humberside,2201 990 0003,24Hr,21,180
T0,02JAN,9:00a,Humberside,2201 990 0003,24Hr,10,90
T0,03JAN,3:40p,West Aedale,0022 760 3303,24Hr,6,50
T0,04JAN,1:00p,Wednesfield,2201 990 0003,24Hr,12,110
T0,05JAN,5:00p,Oxley,2201 990 0003,24Hr,8,70
T1,01FEB,10:13a,Frankfurt,+49 621 41 12 15,24Hr,23,320
T1,02FEB,1:40p,Madrid,+34 515 14 13 12,24Hr,6,99
T1,03FEB,11:04a,Paris,+33 121 31 41 51,24Hr,10,150
T1,04FEB,7:32p,Rome,+39 131 21 51 14,24Hr,16,210
T1,05FEB,10:27p,Zurich,+41 131 21 71 18,24Hr,12,180
T1,06JAN,10:13a,Frankfurt,+49 621 41 12 15,24Hr,23,320
T1,07FEB,1:40p,Madrid,+34 515 14 13 12,24Hr,6,99
T1,08FEB,11:04a,Paris,+33 121 31 41 51,24Hr,10,150
T1,09FEB,7:32p,Rome,+39 131 21 51 14,24Hr,16,210
T1,10FEB,10:27p,Zurich,+41 131 21 71 18,24Hr,12,180
H0,Ricky,Ricky Guy,611 Stafford Road,Pendeford,Wolverhampton,West Midlands,ST11 3JJ
H1,M2,2381 763 8097,7914 3657 8821 1967,04 March 2015,26MAR15,26 March 2015,25JAN -
26FEB,JAN15-FEB15,004895,-004895,000000
H2,004595,003999,000000,000572,000921,010087,00300,01200,00234,00912,01146,000126,000073,00013
3,000332,Evolvcom 500H,003667,,
T0,01JAN,5:00p,Oxley,2201 990 0003,24Hr,8,70
T0,02JAN,9:00a,Humberside,2201 990 0003,24Hr,10,90
```

For more information on using FreeFlow VI Design Express to generate transactional documents using prefixed delimited transactional data, refer to the *Prefixed Delimited Transactional Data* section of the *FreeFlow VI Design Express User Guide*.

## Linking the Data with the JDT, DBM, or XJT

To create the link between the data file and the JDT, DBM, or XJT file, use one of these methods:

- Use the application program to produce a **STARTLM** or **STARTDBM** or **STARTXML** sequence at the top of the data file.
- Use the job or print submission process, for example, JCL or the print server task to:
  - parse the print file data or print file attributes to determine the appropriate JDT, DBM, or XJT to use
  - concatenate a **STARTLM** or **STARTDBM** or **STARTXML** sequence at the top of the print file
- Use the print submission process to create a submission file that references both the print file (SETLMFILE) and the JDT (**STARTLM**), or DBM (**STARTDBM**), or XJT (**STARTXML**) and forwards it to the printer rather than the print file.
- Use the XGFNub or Virtual printer setting available on DocuPrint NPS or FreeFlow Print Server systems to start the printer using the appropriate JDT, DBM, or XJT file. This allows just the data file without any special VIPP® start commands to be sent directly to the printer.

## Record Structures

The VIPP® language supports these record structures:

Record length	Either variable or fixed length
Vertical control	Through control codes or Printer Control Characters (PCC)
Line printer and field delimited data	
Prefix	Either non-prefixed or prefixed data

### RECORD LENGTH

Records can have either a variable or fixed length.

A record with variable length format is a succession of bytes ending with a valid record delimiter such as a LF, CR, or CRLF. This is the default. Maximum record length is provided by the **SETBUFSIZE** command.

A record with fixed length format consists of a fixed number of bytes regardless of the content. Record size is provided by the **SETBUFSIZE** command (F) option.



Note: Variable and fixed length records cannot be mixed within a job.

### VERTICAL CONTROL

Vertical spacing is controlled by control codes or by Printer Control Characters (PCC).

Control codes control vertical spacing with specific control characters (such as CR, LF, or FF) in the data stream. This is the default.

Printer Control Characters (PCC) inserted as the first byte of each record will control vertical spacing. Refer to **BEGINPCC** and **SETPCC** for PCC definition and enabling information.

Keep this information in mind:

- Vertical control mechanisms are exclusive. When PCC is used, FF are processed as print data. CR and LF are processed as record delimiters but do not influence the vertical spacing. For this reason CR and LF cannot be used as PCC values for variable length records as they are first interpreted as record delimiters.
- On fixed length records, CR and LF are processed as print data. When no PCC is used each record is assigned a print-and-space spacing.
- On variable length records with no PCC, the sequences CR, LF, and CRLF are all equivalent to print- and-space. Specific overprint processing (single CR) can be enabled using the **OVERPRINT\_on** command.
- When no PCC is used, an FF occurring in the middle of a record causes the record to be split in 2 parts. This behavior (as well as the value of the FF sequence) can be changed by the **SETPBRK** command.
- Advanced vertical control is also available through the **SETSKIP** command. Refer to **SETSKIP** for more information.
- Vertical control only applies to line mode. In database mode vertical control is performed by native mode commands in the DBM.

## LINE PRINTER AND FIELD DELIMITED DATA

Data can be formatted either as line printer or as field delimited data.

When formatted as line printer data, each record is a line of data formatted with a line printer presentation. Print files that consist of line printer records are intended for printing on impact printers. This format is intended for use with the Listing or RPE presentation options.

Field delimited records consist of a succession of fields separated by a predefined field delimiter. In general, these files are produced by a database extraction. This format is intended for use with database mode or the Line mode RPE presentation option.

Field delimited is a compact format provided for jobs with a finite number of data items per record, such as mailings, insurance contracts, and registration files.

## PREFIX

A prefixed record consists of a prefix followed by data, line printer, or field delimited related to the prefix. This format is intended for use with the RPE presentation option, which allows you to locate and place each field in the record.

Prefix is a compact format that does not contain unnecessary spaces. This format is provided for jobs such as invoices or bank statements that have an indeterminate number of data items per page.

## Record Structures and Mode/Presentation Relationship

This table summarizes the possible associations of the record structures with modes and presentation options. The shaded areas in the table reflect the most common associations. Other associations can be used for documentation or debugging purposes.

RECORD TYPE	MODE/PRESENTATION OPTION <sup>1</sup>			
	LISTING	RPE	DBM	XML
<b>Line printer</b>	VL / no PCC	VL / no PCC	VL / no PCC	N/A
	VL/PCC	VL/PCC	FL / no PCC	
	FL / no PCC	FL / no PCC		
	FL/PCC	FL/PCC		
<b>Prefix</b>	VL / no PCC	VL / no PCC	VL / no PCC	N/A
	VL/PCC	VL/PCC	FL/no PCC	
	FL / no PCC	FL / no PCC		
	FL/PCC	FL/PCC		
<b>Field delimited</b>	VL / no PCC	VL / no PCC	VL / no PCC	N/A
	VL/PCC	VL/PCC	FL / no PCC	
	FL / no PCC	FL / no PCC		
	FL/PCC	FL/PCC		
<b>XML</b>	VL / no PCC	N/A	N/A	XJT

---

1. VL = variable length FL = fixed length

## Functions

VIPP® functions include:

- Orientation settings, including:
  - portrait
  - landscape
  - inverse portrait
  - inverse landscape
- Font selection, including:
  - horizontal and vertical independent scaling
  - outline
  - background (underline, shading, etc.)
  - kerning
- Font re-encoding
- Absolute placement coordinates
- Text alignment, including:
  - left
  - right
  - center
  - justified
  - rotation to any angle
- Line spacing
- Form and back form enabling for use with cycle forms, multiple levels (stacked forms), and the copy sensitive option
- Variable data storage in a file for later use.
- Segment call, refer to [VIPP resources](#) for more information
- Image call with scaling, alignment, and rotation options for TIFF, EPS, and JPEG files
- Print multiple TIFF, RDO, and Decomposed files as a booklet and print TIFF files that contain multiple pages as a single document
- Box, circle, and polygon drawing with the filled or outlined options
- Media selection with the cycle media and copy sensitive options, and ability to insert slipsheets in the middle of a job
- Color or gray level settings for text and images
- Cyclecopy, based on page, set, or document settings, as well as copy sensitive options
- Multi-Up printing

- Automatic page numbering
- Conditional processing

<b>eq</b>	equal
<b>ne</b>	not equal
<b>gt</b>	greater than
<b>ge</b>	greater than or equal
<b>lt</b>	less than
<b>le</b>	less than or equal
<b>CIEQ</b>	Case Insensitive equal
<b>CINE</b>	Case Insensitive not equal
<b>HOLD</b>	searches for a string anywhere within the selected field or defined area or characters on a line.

These functions were originally intended for use in line mode jobs:

- Margin settings
- Grid settings such as characters per line and lines per page
- Automatic zebra (green bar), generation, and frame generation
- Processing horizontal tabulations listing presentation only
- Processing Backspace characters listing presentation only

## Conditional Processing

Use conditional processing functions for VIPP® to trigger dynamic variants in the processing and layout of data during a job.

Conditions may be set at the page, record, or field level. For more information, refer to commands like **SETRCD**, **SETPCD**, **IF/ ELSE/ENDIF**, **CASE**, and **RPE** conditional entries.

## Dynamic Boxes

The dynamic box feature allows you to draw boxes dynamically based on the variable data submitted in a VIPP® job. This allows boxes to be correctly drawn around data, regardless of the number of lines included in the box.

In native mode and database mode, this feature can be invoked with the **SAVEPP** and related **SHPOS**, **SVPOS**, **HDISP**, and **VDISP** commands.

In line mode, this feature can be invoked in one of two ways:

- Use the capability of RPE entries to call segments with the **SCALL** command in the align procedure parameter. Segments can be defined within the JDT using XGFRESDEF. Each segment will dynamically draw specific elements of the boxes (top, sides, bottom, etc.). Refer to [Example 6: Bank statement using dynamic boxes and Data Driven Graphics](#), and to the demonstration jobs dynbox3.lm and samddg.lm for further information.
- Using RPEPOS in a procedure defined by ENDPAGE. For more information, refer to the examples of RPEPOS in the *VIPP® Language Reference Manual*.

In any mode, Background Attributes (BATkey) can be used to highlight specific records or fields individually. Refer to **SETBAT**, **SETTXB**, and **INDEXBAT**.

## Data Driven Graphics

Data Driven Graphics (DDG) also known as Business Graphics, allow you to configure and print bar charts, pie charts, and curve charts. VIPP® **DDG** commands, **DRAWPIE**, **DRAWBAR**, and **DRAWCRV** are described in the *VIPP® Language Reference Manual*.

Examples of DDG coding in native mode are provided in the file `/usr/xgf/demo/samddg.nm`. Coding examples of DDG in line mode is provided in Example 6 in [VIPP file examples](#) and in the file `/usr/xgf/demo/samddg.lm`. For a complete palette of Data Driven Graphics samples, print the files `/usr/xgf/demo/samddg.ps`.

## Native Mode Prefix

When using line mode, the application may dynamically invoke VIPP® resources or features by embedding Native Mode Prefix (NMP) records in the data stream.

These records are identified by a specific prefix followed by one or more commands processed as VIPP® commands rather than printable data. The default prefix is % % XGF. To change the default use the **SETNMP** command.

This feature allows you to dynamically change layout settings such as forms, media, and JDts on a page-to-page basis, or to directly invoke resources such as logos and images on the page.

When using PCC record format, the PCC byte associated with an NMP must be either Skip to channel one or no space. When no PCC is used, the print-and-space default is ignored for NMP records.

Page dependent NMP records such as SETFORM, SETMEDIA, and SETJDT must immediately follow a Skip to channel one or Form Feed.

Several consecutive NMP records can be coded.

These are some of the native mode commands that can be used in NMPs:

<b>PAGEBRK</b>	produces a new page
<b>SETFORM</b>	calls a form
<b>SETMEDIA</b>	selects a media
<b>MOVETO</b>	positions the current point
<b>ICALL/SCALL</b>	calls an image or segment at the current point



Note: Use of NMP records can affect the independence of data production and data presentation.

## Distribution List

As an alternative to SETCYCLECOPY, use the Distribution List (DL) feature also referred to as set labeling to obtain multiple copies of a document. For example, use this feature to produce a report or publication for distribution to a list of addressees.

**SETDLFILE** associates a DL file and a JDT and produces cover pages at the beginning of each set of the document. These pages are created by processing the DL file using the associated JDT and performing the same process as STARTLM. The number of copies produced is equal to the number of pages in the DL files. **SETCYCLECOPY** cannot be used when using this feature.

**SETDLFILE** can be used in a native mode file, a JDT, or a submission file.

## End of File

VI Compose automatically terminates line mode, database mode, or XML mode and performs a RESET when these situations occur:

- At end of file. End of file can be either physical, when the file is spooled to the DocuPrint NPS and FreeFlow Print Server controllers, or at a timeout when the file is received on a communication channel, such as on a parallel port on a DocuPrint N40.
- When the printer reaches a record that begins with the PostScript end of file indicator ( % % EOF).
- When the printer reaches a record that begins with the start of a new PostScript file indicator ( % ).

To concatenate independent jobs in the same print file, restart line mode or database mode by coding a new **STARTLM/ STARTDBM** command and referencing a new JDT or DBM after the % % EOF record, as illustrated in this example.

```
%!
(job1.jdt) STARTLM
first batch of data formatted according to job1.jdt
.....
last line of batch 1
%%EOF
(job2.jdt) STARTLM
second batch of data formatted according to job2.jdt
.....
last line of batch 2
%%EOF
```

To optimize performance when changing JDTs on a page-by-page basis, use the **SETJDT** command as an NMP as illustrated in this example.

```
.....
end of page n
<FF>%%XGF (xyz.jdt) SETJDT
start of page n+1
....
```

<FF> in the above example indicates the form feed character in the data file.



Note: Use the % % EOF string to terminate a line mode or database mode file when the file is received on a communication channel. If you fail to do so, the job will end correctly only when a timeout occurs or when the next job is a VIPP® or PostScript job that starts with a % ! sequence. Any other scenario can cause unpredictable results.

## Coding Examples

This section contains VIPP® coding examples for these modes:

- Native mode coding
- Line mode file
- Job Descriptor Ticket
- Database file
- Data Base Master file

You can find additional coding examples in [VIPP file examples](#).

### NATIVE MODE CODING

This is an example of VIPP® coding in line mode.

```
%!
LAND                                % Set landscape orientation
(TBC12.frm) SETFORM                % Set form to be used for the document
175 2292 MOVETO                     % Set current print position
(TBClogo.tif) 1 0 ICALL             % Print a TIFF image at that location
/NHEB 18 SETFONT                    % Set current font
720 2240 MOVETO                     % Set new current print position
(Trade Bank Corp.) SHC              % Print text centered at current pos.
*****
PAGEBRK                             % End of page
```

### LINE MODE FILE

This is an example of VIPP® native mode coding.

```
%!
(r4.jdt) STARTLM
SALES REPORT - Europe
Product Division      France  U.Kingdom  Germany  Spain   Total
...
BUSINESS DIVISIONS
Personal Document Products 230.120  340.200  190.200  210.000  2.234.000
Office Document Systems    210.000  540.000  200.400  123.040  1.200.000
Office Document Products   210.000  540.000  200.400  123.040  1.200.000
Xsoft                      230.120  340.200  190.200  210.000  2.234.000
...
%%EOF
```

### JOB DESCRIPTOR TICKET

This is an example of VIPP® coding of a JDT.

```

%!
%%Title: r4.jdt
%%CreationDate: may 93
%%Creator: JYB/RXCH
%%Copyright: (C) 1993 by Rank Xerox AG (RXCH). All right reserved.
%-----
% set orientation
%-----
ILAND  % PORT - Portrait   IPORT - Inverse Portrait
% LAND - Landscape  ILAND - Inverse Landscape
%-----
% set forms and media (1 copy)
%-----
2 SETMAXFORM
(xgfts3.frm) 0 SETFORM
(rxlogo.frm) 1 SETFORM
%-----
% set RPE fonts
%-----
/F1 /NHEB  9 13 INDEXFONT
/F2 /NHEB 10 15 INDEXFONT
/F3 /NHEB 13 15 INDEXFONT
/F4 /NHEB 22 18 INDEXFONT
/F5 /NHEB 13   INDEXFONT
/F6 /NHEB 60 20 INDEXFONT
%-----
% RPE definition
%-----
4 BEGINRPE
% Almt. rot.  Xinit  Xdispl  Yinit  Ydisp  Rec.pos.  Length  Font  Color
1 FROMLINE
[ 2  0      838      0      295      0      00      99    /F4    BLACK ]
[ 2  0      833      0      300      0      00      99    /F4    DARK  ]
[ 2 -90     3370      0     1240      0      14      99    /F6    DARK  ]
2 FROMLINE
[ 2  0      615      0      445      0      00      33    /F1    WHITE ]
[ 2  0     1200      0      445      0      33      12    /F1    WHITE ]
[ 2  0     1520      0      445      0      45      12    /F1    WHITE ]
[ 2  0     1830      0      445      0      57      12    /F1    WHITE ]
[ 2  0     2140      0      445      0      69      12    /F1    WHITE ]
[ 2  0     2462      0      445      0      81      12    /F5    WHITE ]
3 FROMLINE
[ 0  0      230      0      560      75     00      33    /F2    BLACK ]
[ 1  0     1345      0      560      75     33      12    /F2    BLACK ]
[ 1  0     1658      0      560      75     45      12    /F2    BLACK ]
[ 1  0     1976      0      560      75     57      12    /F2    BLACK ]
[ 1  0     2286      0      560      75     69      12    /F2    BLACK ]
[ 1  0     2610      0      560      75     81      12    /F3    BLACK ]
ENDRPE

```

## DATABASE FILE

This is an example of VIPP® coding of a database file.

```

%!
(1etter.dbm) STARTDBM
CHER:TITRE:FNAME:NAME:ADRESS1:ADRESS2:ZIP:CITY
Cher:Monsieur:Alain:DUPONT:3, rue de la gare::1200:Genève
Chère:Madame:Martine:BELLEGAUGE:chez Lulu:12, place Grenu:1034:Lausanne
*****
%%EOF

```

## DATA BASE MASTER FILE

This is an example of VIPP® coding of a DBM file.

```
%!
%%Title: letter.dbm
DOT3 SETUNIT
PORT
/M /NTMR 12 11.3 INDEXFONT      % M = medium font
/B /NTMB 12 11.3 INDEXFONT      % B = bold font
/D BLACK INDEXCOLOR             % D = Dark
/H BLUE INDEXCOLOR              % H = Highlight color
1200 0 360 0 SETMARGIN
48 SETLSP
360 3200 MOVETO
H (rxlogo.tif) 1 0 ICALL D
1500 2850 MOVETO
M ($$TITRE. $$FNAME. //B$$NAME.) VSUB 0 SHMF M
ADRESS1 SHL
ADRESS2 SHL
ZIP SH ( - ) SH CITY SH
1780 SETCOLWIDTH
360 2300 MOVETO
($$CHER. $$TITRE. $$NAME.,) VSUB SHL
NL
(Die welt, in der Sie leben, //H$$CHER. $$TITRE. $$NAME.//D, ist - in
Ubereinstimmung mit Ihrer Persönlichkeit - aussergewöhnlich.) VSUB 3 SHP NL
(So aussergewöhnlich wie der Trembley Brooklyn, ein Automobil der
exklusivsten Klasse.wobei sich der Begriff Exklusivitat sowohl auf
das aussergewöhnlich hohe Niveau der) 3 SHP NL
(Verarbeitung, des Luxus, als auch auf die Sicherheit und die Fahrleistungen erstreckt.) 3 SHP
NL
(Lassen Sie sich überraschen. wir werden Ihnen die Aufmerksamkeit widmen,
die Sie verdienen.) 3 SHP NL
(Gerne erwarten wir Ihre Antwort und verbleiben inzwischen mit freundlichen
Grussen.) 3 SHP NL
H (signa.tif) .75 0 ICALL D
NL NL NL NL
(Your W. Signature) SHL
(MANAGING DIRECTOR) SHL
PAGEBRK
```

## XML MODE CODING

This is an example of VIPP® coding in XML mode.

```

%!
%%Title: list.xjt

ORITL
[[ 200 420 2100 2800 0 ]] SETLKf
64 SETLSP
(Page #) 1 2300 220 6 SETPAGENUMBER
/C1 BLACK INDEXCOLOR
/C2 RED INDEXCOLOR
/C3 BLUE INDEXCOLOR
/F1 /NHE 12 INDEXFONT
/F2 /NHEB 13 INDEXFONT
/F3 /NHEN 12 INDEXFONT
/F4 /NHEB 18 INDEXFONT

/B0 null INDEXBAT
/B1 /CH_XLTG INDEXBAT
/B2 /PB_XLTG INDEXBAT
/B3 /SB_LMEDG INDEXBAT

{ ORITL
  200 210 MOVETO
  F4 (Customer List) SH
  NL NL NL
  2100 SETCOLWIDTH
  F1 B1
    0 MOVEHR (Title) SH B0
    120 MOVEHR (First Name) SH
    430 MOVEHR (Last Name) SH
    800 MOVEHR F3 (Street/Post code/City/Country) SH
    1500 MOVEHR F1 (Birthday/Phone/Email) SH
} SETFORM

BEGINXPD
BTA /Customer
{
  { IF FRCOUNT 0 ne {NL} ENDIF
    0 MOVEHR F1 ^USER_TITLE SH
    120 MOVEHR ^USER_FIRST CASETI SH
    430 MOVEHR F2 ^USER_LAST SH
    800 MOVEHR F1 ^USER_STR SH
    1500 MOVEHR ([^USER_BIRTH_DD=].[^USER_BIRTH_MM=].[^USER_BIRTH_YY=])
  B SH
    NL
    800 MOVEHR ^USER_POSTCODE SH
    1150 MOVEHR ^USER_CITY SH
    1500 MOVEHR ^USER_PHONE SH
    NL
    800 MOVEHR ^USER_COUNTRY SH
    1500 MOVEHR ^USER_EMAIL CASELOW 600 100 SHmf
    40 NL
    0 40 2100 0 G_S2 DRAWB
  } ETA
} ENDXPD

```

VSU

## XML

This is a sample of an XML data file.

```

<?xml version="1.0" encoding="ISO-8859-1" ?>
- <CustomerInfo>
- <Customer>
  <USER_ID>10144</USER_ID>
  <USER_TITLE>Mr</USER_TITLE>
  <USER_FIRST>Pravin</USER_FIRST>
  <USER_LAST>Chirac</USER_LAST>
  <USER_STR>38 Launde Road</USER_STR>
  <USER_POSTCODE>LE2 4HG</USER_POSTCODE>
  <USER_CITY>Leicester</USER_CITY>
  <USER_COUNTRY>GBR</USER_COUNTRY>
  <USER_PHONE>0116 2710581</USER_PHONE>
  <USER_EMAIL>pravin.chirac@powerconv.alstom.com</USER_EMAIL>
  <USER_BIRTH_DD>31</USER_BIRTH_DD>
  <USER_BIRTH_MM>3</USER_BIRTH_MM>
  <USER_BIRTH_YY>1961</USER_BIRTH_YY>
  <USER_COMPUTER>yes</USER_COMPUTER>
  <USER_OPERATINGSYS>win95</USER_OPERATINGSYS>
  <USER_CAMERA />
  <USER_MODEL />
  <USER_CONNECTION>Memory stick/Floppy Disk Adapter</USER_CONNECTION>
  <USER_SONYID>3193750702002196</USER_SONYID>
  <USER_SDMENO />
</Customer>
- <Customer>
  <USER_ID>10154</USER_ID>
  <USER_TITLE>Mr</USER_TITLE>
  <USER_FIRST>JAMES</USER_FIRST>
  <USER_LAST>BLAIR</USER_LAST>
  <USER_STR>PJ</USER_STR>
  <USER_POSTCODE>RG8 8LA</USER_POSTCODE>
  <USER_CITY>READING</USER_CITY>
  <USER_COUNTRY>GBR</USER_COUNTRY>
  <USER_PHONE>07970260802</USER_PHONE>
  <USER_EMAIL>JAMES777_99@YAHOO.COM</USER_EMAIL>
  <USER_BIRTH_DD>31</USER_BIRTH_DD>
  <USER_BIRTH_MM>12</USER_BIRTH_MM>
  <USER_BIRTH_YY>1985</USER_BIRTH_YY>
  <USER_COMPUTER>yes</USER_COMPUTER>
  <USER_OPERATINGSYS>win98</USER_OPERATINGSYS>
  <USER_CAMERA />
  <USER_MODEL>BERKSHIRE</USER_MODEL>
  <USER_CONNECTION>USB cable</USER_CONNECTION>
  <USER_SONYID>8386650604002896</USER_SONYID>
  <USER_SDMENO />
</Customer>

</CustomerInfo>

```

# VIPP® Resources

This chapter contains:

Resource Types ..... 102

Resource Access and Management ..... 113

Resources, files stored in the VIPP® libraries, are used for accessing and processing VIPP® jobs. Information on the different types of resources, the different ways in which VIPP® can access these resources, and the mechanism and syntax used to embed resources in the data stream can be found in these sections:

- Resource types
- Resource access and management

## Resource Types

VI Compose supports these types of resources:

- VI Project
- VI Compose and setup files
- Fonts
- Forms
- Segments
- Images
- Job Descriptor Tickets
- XML Job Tickets
- Data Base Masters
- Distribution lists
- Text files
- FreeFlow Makeready RDO files



Note: VI Design Pro includes a utility to convert PDF and Word files to PostScript resources, which can be accepted by VI Compose.

### VI PROJECT

Processing variable data is the core functionality of the VIPP® programming language. The model used by VI Compose to compose documents is known as Dynamic Document Construction (DDC). DDC is based on real time composition of the document at the printer or imaging device.

In this model, document components are often not part of the submission file that triggers the beginning of the composition process. Components such as images, fonts, and forms can be stored locally on disk drives or accessed from network disk drives.

DDC is highly effective for jobs targeted by VI Compose because it is based on, and supports, the model commonly used for the creation of most documents in production environments where job components are created by several different people or departments. The DDC model also improves performance by enabling a *RIP once / Use many* capability. In short, this functionality allows you to reuse various components of a document with other documents.

VI Project provides a mechanism that simplifies the handling of the job components by logically and physically grouping components as part of a single entity. The logical grouping is the VI Project. The physical grouping is referred to as the VI Project Container.

The files that make up the project as a single entity can be tracked and stored using VI Project. In addition, VI Projects can be used to identify, organize, and store the components of a job under a single name of the project and can also group jobs by categories of the folder. This allows you to package all of the project components in a single file or in the container as well as to use and transfer them among applications, devices, and locations.

## VI COMPOSE AND SETUP FILES

VI Compose setup files are located in the `/usr/xgf/src` directory and are called by the link file. Depending on the capability of the device, these files are executed only once at power up, or once per job. The `/usr/xgf/src/xgf` file for the Sun platform and the `C:\Program Files (x86)\Xerox\VIPP\xgf\src\xgfdos` link file execute the `/usr/xgf/src/xgfunix.run` file on DocuPrint NPS, and FreeFlow Print Server controllers, or the `\xgf\src\xgfdos.run` file on DOS systems. The resource paths are coded in `xgfunix.run` and `xgfdos.run`.

<code>xgf</code> or <code>xgfdos</code>	link file
<code>xgf.lic</code>	license file
<code>xgf.eeh</code>	VI Compose code
<code>xgf.ddg</code>	Data Driven Graphics module
<code>xgf.def</code>	general defaults
<code>xgf.bat</code>	standard background attribute keys (BATkeys)
<code>xgf.mup</code>	standard Multi-Up definitions
<code>xgf.pcc</code>	standard PCC definitions
<code>xgfdos.run</code>	DOS core files and resource library paths
<code>xgfunix.run</code>	UNIX core files and resource library paths
<code>xgf.417</code>	PDF417 barcode
<code>xgf.dmx</code>	DATAMATRIX barcode
<code>xgf.max</code>	MAXICODE barcode
<code>fontlist</code>	Latin font list
<code>nullfl</code>	special font list symbol and dingbats
<code>vipp.vsm</code>	VIPP® Services
<code>sucoat.cck/scoat.cck</code>	Custom colors
<code>xgf.gep</code>	standard Colorkeys, Pattern keys, and Graphic Element Property keys (gep keys)
<code>chk.def</code>	Chinese, Japanese, Korean (CJK) font encoding, mapping, and character tables.
<code>vipp.spi</code>	Fluorescent and Infrared color enablement
<code>xgf.qrc</code>	QRCODE barcode
<code>arb.def</code>	Arabic language definition file
<code>xgf.4cb</code>	4-State barcode

## Fonts

### Printer (PS) font support

VI Compose supports Printer (PS) fonts either as PostScript resources or as VIPP® resources.

- As PostScript resources, fonts must be accessible to the PostScript interpreter on which VIPP® is running. VI Compose does not perform any resource management functions on these fonts. It relies on the standard PostScript mechanism to locate fonts and accesses them using their PostScript font name.

Use this method for the initial font set fonts that are pre-loaded in the PostScript device, and for any additional fonts loaded on the device that will be available to PostScript applications. Fonts embedded in PostScript files, like EPS, are also supported as PostScript resources.

Loading fonts as PostScript resources is device-specific. Refer to the appropriate device manual for instructions on how to load additional fonts onto a specific device. For example DP/NPS printers use the *Install Font* command and FreeFlow Print Server systems use the Font Manager. VIPP® Manage can be used to load fonts on VIPP®-enabled desktop or office print devices.



Note: For printing systems that do not allow direct access to the printer operating system, such as Xerox Office products, you must use a third party VIPP® design tool that supports remote resource management to load fonts. For a complete list of third party VIPP® design tools that support remote resource management functions contact a local Xerox representative.

- As VIPP® resources, font files must be loaded in one of the VI Compose directories defined by SETMPATH. The recommended directory is `xgfc/fontlib`. When VI Compose initially accesses the VIPP® resources the font file name is used as an operand to SETENCODING. During subsequent access, VI Compose uses the VIPP® font name. The Variable Information Suite applications also perform resource management functions such as embedding the font in a VIPP® self-contained print stream.

Use this method for fonts that are specific to a VIPP® job or VI Project, rather than to a specific device. This method ensures that the font will always be included with the project and automatically embedded in self-contained print streams. It also provides a method to install additional fonts in a VIPP® environment and avoids the need to install the fonts on each target device.

VI Compose supports any font type Type 1, Type 3, and composite supported by the PostScript interpreter.



Note: VI Specialty Imaging fonts such as GlossMark, MicroText, and Correlation can be installed either as printer (PS) fonts or as VIPP® resources.

### OTF/TTF font support

Support of OTF/TTF fonts is PS interpreter dependant; some PS interpreters do support OTF/TTF fonts. If the PS interpreter on which VIC is installed supports OTF/TTF fonts then they can be used in VIPP® code.

An OTF or TTF font can be placed in the same directory where PostScript fonts normally reside. The .otf or .tff extension must be removed when copying the OTF/TTF file to the font directory. After the fonts have been copied into the font directory, they can be referenced from within the VIPP® code by using the font file name, not the internal PostScript font name /FontName.

#### Example:

Copy `C:\WINDOWS\Fonts\RosewoodStd-Regular.otf`

to:

- `x:\vide\fonts\RosewoodStd-Regular` for VDP

- x:\vviewer\fonts\RosewoodStd-Regular for VIE
- /opt/XRXnps/resources/ps/fonts/RosewoodStd-Regular for FFPS
- x:\efi\server\adobe\fonts\RosewoodStd-Regular for EFI
- x:\CXP6000\General\RIP\fonts\RosewoodStd-Regular

- x:\WINDOWS\Fonts for ViE

Then use the font as a PostScript font in the VIPP® code by using the font file name as in one of the following Options:

**Option 1 (if no re-encoding is needed):**

```
/RosewoodStd-Regular 20 SETFONT
```

**Option 2 (if re-encoding is needed):**

using SETENCODING:

```
[ /NRWSR /RosewoodStd-Regular ] (sun8) SETENCODING
```

or by adding the following line to encoding/fontlist:

```
/NRWSR /RosewoodStd-Regular
```

and select the font with the VIPP® font name: /NRWSR 20 SETFONT

**Other fonts**

Refer to [Getting started](#) for information about purchasing and downloading additional fonts, such as barcode and multi-byte fonts. Fonts purchased from other sources may have nonstandard encoding tables, which can cause incorrect characters to print. Custom encoding can be used to resolve this problem. For further information, refer to [VIPP font lists and encoding tables](#).



**Caution:** Fonts are often subject to licensing agreements and must be used in accordance with the terms of the agreement. You are responsible to check the licensing agreement and comply with the terms before installing fonts on the printer, in the fontlib, or in project directories.

**Font encoding**

A PostScript font is a collection of characters or character set associated with a specific font name. Each character in a font contains an outline description or vector or a bitmap that represents the character, and a PostScript-assigned name for each character. However, data streams call out each character using a byte value. For this reason a PostScript font contains an encoding table matching the byte value with the character name.

This table shows the relationship between the hex value in the data, the PostScript character name, and the actual printed character using the standard ASCII encoding table.

HEX VALUE	POSTSCRIPT NAME	CHARACTER
26	/ampersand	&
20	/space	
24	/dollar	\$
2A	/asterisk	*
41	/A	A
32	/2	2
7A	/z	z



Note: Although a character set in a font can be relatively large, byte values are in the range of 0-255; therefore, the length of the encoding table is limited to 256. This means that only a subset of 256 characters is available with an encoding table.

### VIPP® font lists and encoding tables

Two VIPP® commands can be used to select fonts, SETFONT and INDEXFONT. Any font available to the interpreter can be selected by these commands using the original PostScript font name. For example:

```
/Helvetica 12 SETFONT
```

PostScript fonts are built with a standard encoding table that covers only a small subset of the character set. Latin accented characters are not included. When the original encoding table does not include a specific character, or a character is assigned to a byte value that does not match the data stream, the encoding table must be altered. This process is called *font re-encoding*.

VIPP® provides a font re-encoding mechanism through the SETENCODING command. SETENCODING:

- Assigns a new name generally shorter to the font

- Alters the encoding table

To re-encode fonts, provide a font list and an encoding table to SETENCODING.

### Font lists

VI Compose assigns a short name to each PostScript font in the initial font set as shown in this sample. Fonts can then be selected using this VIPP® short name.

% VIPP® NAME	POSTSCRIPT NAME
/NHE	/Helvetica
/NHEB	/Helvetica-Bold
/NHEO	/Helvetica-Oblique

These two pre-defined font lists are stored in the `xgf/encoding` directory. Other font lists can be created to accommodate special requirements.

- font- list contains all Latin fonts and is associated with an encoding table (pcsun) that includes most latin accented characters used on UNIX, Windows, and DOS systems.
- nullfl contains special fonts such as symbol and dingbats. These fonts are not re-encoded, they are only assigned a shorter name.

Fonts can be added to one of the pre-defined font lists when they are added to the system. Use fontlist to re-encode the fonts using latin encoding and nullfl to assign a short name only.



Note: Do not place non-Latin character fonts such as barcode and postnet in fontlist, they will be unusable.

VI Compose can call out fonts that are not included in the font lists by using the original PostScript font name. The PostScript font name can be found in the documentation provided with the font, or by searching for the string `/FontName` in the PostScript font file.



Note: On FreeFlow Print Server fontlist and nullfl contain an expanded list of PostScript fonts. When updating to a new version of VI Compose, old possibly customized files in `xgf/encoding` will be saved in a directory called `xgf/encoding.xxx`, which contains a backup of all VIPP® encoding files.

### Encoding tables

VI Compose provides these predefined encoding tables:

- sun8
- pc8
- pcsun
- ebcdic
- mac

- utf8

### Creating custom font lists and encoding tables

Custom font lists and encoding tables can be created to meet special requirements. These must be stored in one of the libraries referenced by SETEPATH or SETPPATH in project mode; `xgfc/encoding` is the preferred directory.

Follow these steps to create a custom re-encoding table:

1. Create a unique font list file. Save this file using the .lst extension. This is an example of a font list file:

```
%!
%% Title: Myencoding.lst

% New Name      PSfont name
/Myfont1        /Times-Roman      % PS resource
/Myfont2        /Helvetica        % PS resource
/Myfont3        (corporatefont.pfb) % VIPP resource
```

#### Where:

New Name                                  is the new font name  
 PS font name                                is the original font name or font file name

2. Create the encoding file. The encoding file contains a list of all the values to be changed or re-encoded. Any values not included in this list match the default PostScript ASCII encoding table or Standard Encoding. Save this file using the .enc extension.

This example illustrates how to re-encode the byte value hex 7E, the tilde character in ASCII to the space character:

```
%!
%% Title: Myencoding.enc

% Incoming Hex value - PostScript Character Name
16#7E                        /space
```

3. Activate re-encoding by adding the SETENCODING command to the JDT file as follows:  
 (Myencoding.lst) (Myencoding.enc) SETENCODING
4. Select the re-encoded fonts using the SETFONT or INDEXFONT commands as follows:

```
/Myfont1 12 SETFONT.
/F2 /Myfont2 14 INDEXFONT
```

### In-line coding

An alternative to font re-encoding is to use in-line coding to place all definitions in the JDT. This example will produce the same result as steps 1 -3 above:

```
/Myencoding { 16#7E /space } XGFRESDEF
[ /Myfont1    /Times-Roman
  /Myfont2    /Helvetica
  /Myfont3    (corporatefont.pfb)
] (Myencoding) SETENCODING
```

## FORMS

A form is a single-page VIPP® native mode or PostScript document intended to be reproduced identically in the background on each page of a job. Forms can be invoked using either SETFORM or SETBFORM.

When directly invoked by SETFORM or SETBFORM, forms must be coded in VIPP® native mode or simple PostScript code that does not use the *currentfile* operator and encapsulated between braces {}. Adding the command FSHOW after the closing brace allows you to print the form only, and automatically invokes the caching feature. These types of forms must be stored in one of the libraries referenced by SETFPATH. Forms that contain variable references should not be cached.

PDF, PS, EPS, TIFF, and JPEG files created by document processing applications using common drivers for example, .prn files can be invoked directly by using the CACHE command in the SETFORM or SETBFORM syntax. These types of forms must be stored in one of the libraries referenced by SETFPATH, SETIPATH, or SETMPATH.

Multiple levels of forms on a page are supported and controlled by SETMAXFORM. Copy-sensitive forms and cycle forms are also supported.

## SEGMENTS

A segment is a VIPP® native mode or a PostScript fragment intended to be reproduced once or several times at specific locations on a page. You must store a segment in one of the libraries referenced by SETFPATH. Segment can be invoked using either SCALL or FCALL.

PDF, PS, EPS, TIFF, and JPEG files created by document processing applications using common drivers typically .prn files must be invoked using the CACHE command with the SCALL syntax. This type of segment must be stored in one of the libraries referenced by SETFPATH, SETIPATH, or SETMPATH.

## IMAGES

When using an image as a VIPP® resource, it can be a TIFF, JPEG, EPS, PS, or PDF file. A TIFF file is a bitmap stream wrapped in a TIFF header. A JPEG/JFIF file is a JPEG bitstream wrapped in a JFIF header. Images can be reproduced one or more times at specific locations on the page.



Note: PDF files may need to have embedded EPS information stored in the PDF file, refer to [PDF files as resources](#).

Use the ICALL command to invoke an image when it is stored in one of the libraries referenced by SETIPATH. Use the CACHE command with the SCALL syntax to invoke an image when it is stored in one of the libraries referenced by SETIPATH, SETFPATH, or SETMPATH.

VI Compose supports only a subset of TIFF and JPEG files. Refer to ICALL for further information on the supported image files.

## PDF FILES AS RESOURCES

Using PDF files as VIPP® resources is now supported when using VI Design Express or VI Design Pro. PDF files or PDF referenced XObjects can print directly to Adobe PDF Print Engines (APPE) using VIPP® or the PDF files will need to be modified by the design tools to embed EPS image information into the PDF files if printing to a PostScript (PS) print device. For more information about using PDF files as VIPP® resources refer to the *FreeFlow VI Design Express User Guide*, or the *FreeFlow VI Design Pro User Guide*.

## JOB DESCRIPTOR TICKETS

A Job Descriptor Ticket (JDT) is a file that contains the VIPP® commands, which control the page composition of a job.

The VIPP® commands contained in the JDT format the data for printing for example, by changing the font, orientation, spacing, or by calling forms or other resources. In general, use JDTs with VIPP® line and XML mode. However, JDTs can also be used with native mode and database mode.

With the exception of Native Mode Prefix (NMP) records, the print file on which a JDT is applied is a file that consists of line printer or raw data prefixed records without embedded page composition commands. A JDT must be stored in one of the libraries referenced by SETJPATH. Invoke JDTs using STARTLM or SETJDT.

## XML JOB TICKETS

An XML Job Ticket (XJT) is a file that contains instructions on how to process and arrange XML data into a document. It is similar to a line mode JDT. It contains all global layout definitions for the document like orientation, forms, media, frames, fonts, colors, and so on plus an XML Processing Definition (XPD) table that describes specific actions to be performed on specific XML tags.

The XJT must be provided as an operand to STARTXML.

## DATA BASE MASTERS

A Data Base Master (DBM) is a file that contains VIPP® native mode commands and variable references used to control the page composition of a job. The print file on which a DBM is applied is a file that consists of field delimited records produced by a database extraction. You must store DBMs in one of the libraries referenced by SETFPATH. Invoke DBMs using STARTDBM. For further information, refer to [VIPP data streams](#).



Note: Any database, mainframe, or PC file that can be extracted to field delimited records is a candidate for a VIPP® database mode job.

## DISTRIBUTION LISTS

A distribution list (DL) is a data file on which a specific JDT is applied to produce the cover pages used for the distribution of numerous copies of a document. You must store distribution lists in one of the libraries referenced by SETMPATH.

Use the Distribution list (DL) feature also referred to as set labeling to obtain multiple copies of a document, as an alternative to using the SETCYCLECOPY command. For example, this feature can be used to produce a report or publication for distribution to a list of addressees.

Invoke a distribution list using SETDLFILE. The SETDLFILE command associates a distribution list file with a JDT and produces cover pages at the beginning of each set of the document. These pages are created by processing the DL file using the associated JDT and performing the same process as STARTLM. The number of copies produced is equal to the number of pages in the DL file. SETCYCLECOPY cannot be used when using this feature. Use SETDLFILE in a native mode file, a JDT, or a submission file.

## TEXT FILES

Text files can be used as a VIPP® resource. Use VSUB to incorporate the contents of a text file into a string.

This example will print the contents of a file named text1.txt, which must be located in a directory specified in SETMPATH or SETPPATH:

```
( [=text1.txt=] ) VSUB 0 SHP
```

This example shows how to use a variable field name as input:

```
( $$ [=fieldname=] . ) VSUB 0 SHP
```

The string value contained in the variable or database field is the first substitution, then, assuming that the substituted value is a valid text file (name.txt), the file name will be replaced with contents of the text file.

### FREEFLOW MAKEREADY RDO FILES

Makeready Raster Document Object RDO or .rdo files can be used as VIPP® resources. In order for VI Compose to be able to use these resources, the Makeready library containing the RDO files must be made available to VI Compose. The RDO files may contain VIPP® commands. These commands are placed in the RDO file using the Makeready *VIPP® Text* interface, refer to [VI Compose/Makeready API overview](#) for more information.



Note: VI Compose can only process TIFF-based RDO files.

## Resource Access and Management

Resources used in a VIPP® job, such as forms or segments, may be either external or embedded in the job. When a resource is not embedded, VI Compose tries to locate and load it from the file system available to the PostScript interpreter at the time the job is imaged.

Use of external resources greatly reduces the production time and size of the print job. Because they are not part of the job, external resources must be deployed to the file system of the target device prior to job submission. Production printers and electronic document applications can share a common resource repository on a network file system where resources are deployed only once to a single location. When using desktop and office printing systems that do not allow direct access to the printer operating system you must use VIPP® Manage or a third party product that supports remote resource management to load fonts. For a complete list of third party VIPP® design tools that support remote resource management functions contact a local Xerox representative.

Use of embedded resources ensures print job autonomy. This method is appropriate when the job is sent to a device that is outside the scope of resource deployment or without a file system, such as a diskless desktop printer.

Embedded resources, when present in a job, must be grouped either in the global resource section before the setup section, or in the body section before a page. In any case they must appear in the data stream prior to any invocation by a VIPP® command.

Embedded resources are loaded in memory and their scope is limited to the job. They are cleared from memory at the end of the job.

An embedded resource takes precedence over an external resource with the same name. Test jobs during the development phase can contain embedded resources that temporarily obscure external resources.

VIPP® commands such as SETFORM or SCALL only require the base name of the resource as an operand. This name is used as a key to embed the resource. When the resource is not embedded, VI Compose looks for the external resource in a set of directory paths defined by the SETxPATH commands. This definition is part of the VIPP® configuration resident on the target device. This makes the VIPP® data stream device independent, as the resolution of the resource paths occurs on the device itself.

When VI Compose cannot access a resource because it is neither embedded nor external the job aborts and the error message *VIPP®\_unable\_to\_locate* is reported by the PostScript interpreter along with the resource name.

The XGFRESDEF command describes the various syntaxes used to embed resources depending on their type. The resource name used in these syntaxes is the name used as an operand in VIPP® commands invoking resources.

Consult a Xerox representative for technical documentation about applications and emitters that create full autonomous VIPP® print jobs including VI Compose setup files and VIPP® resources also referred to as *self-contained* job.

### MANAGING VIPP® RESOURCES



Note: All SETEPATH, SETFPATH, SETIPATH, SETJPATH, SETMPATH, and SETPPATH command parameters are set either in the xgfdos.run or xgfunix.run file. Do not attempt to place these commands in a VIPP® job or job portability will be compromised, or in a VI Project both portability and project organization will be compromised.

VIPP® resources can be located on any disk, on another machine, or on a network server. In order for the VIPP®-enabled device to access these resources, provide access to them by performing these steps:

1. Set up a link file, `/usr/xgfc`, that points to the location of the resources. Use the `/usr/xgfc` directory to store the custom files, thus allowing the jobs to continue to run after each upgrade.
2. For all supported DFE's supporting the PostScript Interpreter, edit the `/usr/xgf/src/xgfunix.run` or `x:\xgf\src\xgfdos.run` file to specify alternate locations.
3. Add the necessary paths to the list of directories defined by SETEPATH, SETFPATH, SETIPATH, SETJPATH, and SETMPATH for legacy jobs and SETPPATH for VI Projects.

For related information refer to the individual commands in the *VIPP® Language Reference Manual*, and to **VI Projects** in the *VIPP and VI Compose overview*.

# VI Compose Files and Utilities

This chapter contains:

Link and Setup Files.....	116
Utilities and Demonstration Files .....	125

These VI Compose files and utilities are described:

- Link and setup files
  - xgfunix.run link file (UNIX and Windows FFPS)
  - xgfdos.run link file (VI Windows products)
  - General defaults setup file
- Utilities and demonstration files
  - xgfprint
  - VI Project Container Filter
  - VIPPManage
  - VI Compose version information
  - License Request forms
  - VIPP demonstration files

## Link and Setup Files

Each VI Compose installation includes these editable files:

- **xgfunix.run link file (UNIX and Windows FFPS)**
- **xgfdos.run link file (VI Windows products)**
- **General defaults setup file**

For your convenience the text version of these link and setup files is provided in the sections that follow. Each section includes the location and a brief description of how the file is invoked.



Note:

### Link files vary

The link files in this document may be slightly different from the versions on the system as some paths are changed or added to reflect the system's requirements.

### Print and Forget Directories

A VPC Print and Forget Directory (vpcpfd) is only used for applications that support temporary deployment of VPC files like VIE, VIeC, VDE and VPCF on FFPS. When an application does not support temporary deployment of VPC files, any VPC it opens is automatically permanently deployed in the project folders (likely under xgfc). DOS link files for those applications do not contain references to vpcpfd.

### XGFUNIX.RUN LINK FILE (UNIX AND WINDOWS FFPS)

The link file for UNIX FFPS is provided in `/usr/xgf/src/xgfunix.run`. This file must be invoked by a PostScript run command contained in the initialization file for PostScript interpreters running under a UNIX file system.

```

%!PS-Adobe-2.0
%%Title: FreeFlow VI Compose link file for Unix (xgfunix.run)
%%Creator: JVB
%%CreationDate: Nov 2003
%%Copyright: (C) 1993,1994,1995,1996,1997,1998,1999,2000,2001,2002,2003,2004,
%%+ 2005,2006,2007,2008,2009,2011 Xerox Corporation. All Rights Reserved.
%%Source: $RCSfile: xgfunix.run,v $ $Revision: 1.56 $ - $Date: 2011/05/12 07:27:09 $
%%EndComments

(/usr/xgf/src/xgf.lic) run      % run xgf license
(/usr/xgf/src/xgf.eeh) run     % run xgf prolog
(/usr/xgf/src/vipp.vsm) run    % run VI Compose Services Module

% NOTE: The customer may add new paths or edit the original default paths
% to reflect their local configuration. However, please DO NOT remove
% any of the original default paths.

XGfdict begin                  % push XGfdict on stack

/XGFINI where {pop XGFINI} if  % additional VIPP init %

[ (/usr/xgfc/formlib/)         % customer path for forms & segments
  (/usr/xgf/formlib/)          % demo path for forms & segments
  (/usr/xgf/gshared/)          % demo path for globally shared forms & segments
  (/var/db/forms/)             % path for decomp forms
] SETFPATH

[ (/usr/xgfc/imglib/)          % customer path for images
  (/usr/xgf/imglib/)           % demo path for images
  (/var/db/forms/)             % path for decomp images
  (/var/spool/)                % path for decomp images
  (/usr/xgfc/decomp/)          % path for decomp images
  (/var/spool/XXnps/saved/)     % default path for DocuSP decomp images
] SETIPATH

[ (/usr/xgfc/jdtlib/)          % customer path for jdt
  (/usr/xgf/jdtlib/)           % demo path for jdt
  (/usr/xgf/gshared/)          % demo path for globally shared jdt files
] SETJPATH

[ (/usr/xgfc/encoding/)        % customer path for encoding files
  (/usr/xgf/encoding/)         % demo path for encoding files
] SETEPATH

[ (/usr/xgfc/mislib/)          % customer path for miscell. files
  (/usr/xgfc/fontlib/)         % customer path to font lib
  (/usr/xgf/demo/)             % demo path for demo files
  (/usr/xgfc/formlib/)         % customer path for forms & segments
  (/usr/xgf/formlib/)         % demo path for forms & segments
  (/usr/xgf/encoding/)         % demo path for encoding files
  (/opt/XXnps/resources/ps/mislib/) % project access to DocuSP resource list
  ()                           % null path for full names
] SETMPATH

% NOTE: In SETPPATH, the path containing the directory component "vpcpfd" MUST
% be the FIRST path and no other path in SETPPATH can contain the "vpcpfd"
% component.

```

```
[ (/var/spool/XXRnps/resources/vpcpfd/$$FOLDER./$$PROJECT./) % project path for expanding
'print&forget' VPCs
(/usr/xgfc/$$FOLDER./$$PROJECT./) % project local paths
(/usr/xgfc/$$FOLDER./fshared/) % project folder shared paths
(/usr/xgfc/gshared/) % project global shared paths
(/usr/xgfc/fontlib/) % project access to font lib
(/usr/xgf/encoding/) % project access to standard encoding
(/usr/xgf/gshared/) % project global shared path
(/opt/XXRnps/resources/ps/mislib/) % project access to DocuSP resource list
] SETPPATH

% ALERT! The xgfspool file MUST be writable.
% The following line may have to be edited to specify a path
% to the file xgfspool that is writable.

(/opt/XXRnps/resources/ps/mislib/xgfspool) SPOOLNAME % XGF spoolfile full name

% WARNING!!
% The following statements MUST NOT be moved.
% To do so may cause unpredictable results.
%
% They may be edited only to add or remove a leading comment notation
% (percent sign "%") to enable or disable the indicated features.

(/usr/xgf/src/xgf.gep) run % run xgf predef. color keys and GEP keys
(/usr/xgf/src/scoat.cck) run % run xgf solid coated color keys
(/usr/xgf/src/sucoat.cck) run % run xgf solid uncoated color keys
(/usr/xgf/src/xgf.ddg) run % run data driven graphics
(/usr/xgf/src/xgf.417) run % run PDF417 barcode
(/usr/xgf/src/xgf.max) run % run MaxiCode barcode
(/usr/xgf/src/xgf.dmx) run % run Datamatrix barcode
(/usr/xgf/src/xgf.qrc) run % run QR Code barcode
(/usr/xgf/src/xgf.4cb) run % run USPS4CB barcode
(/usr/xgf/src/cjk.def) run % run cjk default setting
(/usr/xgf/src/arb.def) run % run arb default setting
(/usr/xgf/src/xgf.def) run % run xgf default setting
(/usr/xgf/src/xgf.pcc) run % run xgf PCC definitions
(/usr/xgf/src/xgf.mup) run % run xgf multi-up definitions
(/usr/xgf/src/xgf.bat) run % run predefined BAT keys
(/usr/xgf/src/vipp.spi) run % run Specialty imaging resources
end % mandatory (remove XGFdict from dict stack)
```

The link file for Windows FFPS is provided in % XPS\_HOME2 % /resources/xgf/src. This file must be invoked by a PostScript run command contained in the initialization file for PostScript interpreters running under Windows FFPS..

Depending on the FFPS system, paths will be in the '%os%' format (see xgfunix.run below) or the 'Drive:' format (e.g. D:/Xerox-PS/data/xgfc/mislib/). Be sure to follow the same format when customizing xgfunix.run.

When adding a reference to a mapped drive on FFPS, use only forward slashes '/', not back slashes '\' or double back slashes '\\'. For example, '%os% /Z/Xerox-PS/data/xgfc/mislib/' is acceptable; '%os% \\Z\\Xerox-PS\\data\\xgfc\\jdtlib\\' is not. Note that it is necessary to stop and restart FFPS after creating a mapped drive or FFPS will not recognize it.

UNC paths may also be used. For example, '%os% //11.222.33.44/shared/Xerox-PS/data/xgfc/mislib/' or '//11.222.33.44/shared/Xerox-PS/data/xgfc/mislib/' are acceptable. The advantage of using UNC paths is FFPS does not have to be stopped and restarted to recognize them. Note that you still have to map a network drive to the UNC location to enable access to the location.

```

%!PS-Adobe-2.0
%%Title: FreeFlow VI Interpreter link file for Unix (xgfunix.run)
%%Creator: JYB
%%CreationDate: Nov 2003
%%Copyright: (C) 1993-2016 Xerox Corporation. All Rights Reserved.
%%Source: $RCSfile: xgfunix.run,v $ $Revision: 1.56 $ - $Date: 2011/05/12 07:27:09 $
%%EndComments

(%os%/C/Program Files (x86)/Xerox-PS/resources/xgf/src/xgf.lic) run % run xgf license
(%os%/C/Program Files (x86)/Xerox-PS/resources/xgf/src/xgf.eeh) run % run xgf prolog
(%os%/C/Program Files (x86)/Xerox-PS/resources/xgf/src/vipp.vsm) run % run VI Interpreter
                                                                Services Module

% NOTE: The customer may add new paths or edit the original default paths
%       to reflect their local configuration. However, please DO NOT remove
%       any of the original default paths.

XGFDict begin

[ (%os%/C/ProgramData/Xerox-PS/data/xgfc/formlib/) % customer path for forms &
  (%os%/C/Program Files (x86)/Xerox-PS/resources/xgf/formlib/) % demo path for forms &
  (%os%/C/Program Files (x86)/Xerox-PS/resources/xgf/gshared/) % demo path for globally shared
                                                                forms & segments
] SETFPATH

[ (%os%/C/ProgramData/Xerox-PS/data/xgfc/imglib/) % customer path for images
  (%os%/C/Program Files (x86)/Xerox-PS/resources/xgf/imglib/) % demo path for images
  (%os%/C/ProgramData/Xerox-PS/data/xgfc/decomp/) % path for decomp images
] SETIPATH

[ (%os%/C/ProgramData/Xerox-PS/data/xgfc/jdtlib/) % customer path for jdt
  (%os%/C/Program Files (x86)/Xerox-PS/resources/xgf/jdtlib/) % demo path for jdt
  (%os%/C/Program Files (x86)/Xerox-PS/resources/xgf/gshared/) % demo path for globally shared
                                                                jdt files
] SETJPATH

[ (%os%/C/ProgramData/Xerox-PS/data/xgfc/encoding/) % customer path for encoding
  (%os%/C/Program Files (x86)/Xerox-PS/resources/xgf/encoding/) % demo path for encoding files
] SETEPATH

```

```

[ (%os%/C/ProgramData/Xerox-PS/data/xgfc/mislib/)           % customer path for miscell.
                                                              files
(%os%/C/ProgramData/Xerox-PS/data/xgfc/fontlib/)           % customer path to font lib
(%os%/C/Program Files (x86)/Xerox-PS/resources/xgf/demo/)   % demo path for demo files
(%os%/C/ProgramData/Xerox-PS/data/xgfc/formlib/)            % customer path for forms &
                                                              segments
(%os%/C/Program Files (x86)/Xerox-PS/resources/xgf/formlib/) % demo path for forms &
                                                              segments
(%os%/C/Program Files (x86)/Xerox-PS/resources/xgf/encoding/) % demo path for encoding files
O                                                            % null path for full names
] SETPPATH

% NOTE: In SETPPATH, the path containing the directory component "vpcpfd" MUST
%       be the FIRST path and no other path in SETPPATH can contain the "vpcpfd"
%       component.

[ (%os%/C/ProgramData/Xerox-PS/data/xgfc/vpcpfd/$$FOLDER./$$PROJECT.)/% project path for
                                                              expanding
                                                              'print&forget' VPCs
(%os%/C/ProgramData/Xerox-PS/data/xgfc/$$FOLDER./$$PROJECT.)/ % project local paths
(%os%/C/ProgramData/Xerox-PS/data/xgfc/$$FOLDER./fshared/)    % project folder shared
                                                              paths
(%os%/C/ProgramData/Xerox-PS/data/xgfc/gshared/)              % project global shared
                                                              paths
(%os%/C/ProgramData/Xerox-PS/data/xgfc/fontlib/)              % project access to
                                                              font lib
(%os%/C/Program Files (x86)/Xerox-PS/resources/xgf/encoding/) % project access to
                                                              standard encoding
(%os%/C/Program Files (x86)/Xerox-PS/resources/xgf/gshared/)  % project global shared
                                                              path
] SETPPATH

% ALERT! The xgfspool file MUST be writable. The following line may have to be edited to
%       specify a path to the file xgfspool that is writable.

(%os%/C/ProgramData/Xerox-PS/data/xgfc/temp/xgfspool) SPOOLNAME % XGF spoolfile full name

% WARNING!!
% The following statements MUST NOT be moved. To do so may cause unpredictable results.
% They may be edited only to add or remove a leading comment notation
% (percent sign "%") to enable or disable the indicated features.

(%os%/C/Program Files (x86)/Xerox-PS/resources/xgf/src/xgf.gep) run % run xgf predef. color
                                                              keys and GEP keys
(%os%/C/Program Files (x86)/Xerox-PS/resources/xgf/src/scoat.cck) run % run xgf solid coated
                                                              color keys
(%os%/C/Program Files (x86)/Xerox-PS/resources/xgf/src/sucoat.cck) run % run xgf solid
                                                              uncoated color keys
(%os%/C/Program Files (x86)/Xerox-PS/resources/xgf/src/xgf.ddg) run % run data driven
                                                              graphics
(%os%/C/Program Files (x86)/Xerox-PS/resources/xgf/src/xgf.417) run % run PDF417 barcode
(%os%/C/Program Files (x86)/Xerox-PS/resources/xgf/src/xgf.max) run % run MaxiCode barcode
(%os%/C/Program Files (x86)/Xerox-PS/resources/xgf/src/xgf.dmx) run % run Datamatrix
                                                              barcode
(%os%/C/Program Files (x86)/Xerox-PS/resources/xgf/src/xgf.qrc) run % run QR Code barcode
(%os%/C/Program Files (x86)/Xerox-PS/resources/xgf/src/xgf.4cb) run % run USPS4CB barcode
(%os%/C/Program Files (x86)/Xerox-PS/resources/xgf/src/cjk.def) run % run cjk default
                                                              setting
(%os%/C/Program Files (x86)/Xerox-PS/resources/xgf/src/arb.def) run % run arb default
                                                              setting
(%os%/C/Program Files (x86)/Xerox-PS/resources/xgf/src/xgf.def) run % run xgf default
                                                              setting
(%os%/C/Program Files (x86)/Xerox-PS/resources/xgf/src/xgf.pcc) run % run xgf PCC
                                                              definitions
(%os%/C/Program Files (x86)/Xerox-PS/resources/xgf/src/xgf.mup) run % run xgf multi-up
                                                              definitions
(%os%/C/Program Files (x86)/Xerox-PS/resources/xgf/src/xgf.bat) run % run predefined BAT
                                                              keys
(%os%/C/Program Files (x86)/Xerox-PS/resources/xgf/src/vipp.spi) run % run specialty
                                                              imaging resources
end % mandatory (remove XGFdict from dict stack)

```

## XGFDOS.RUN LINK FILE (VI WINDOWS PRODUCTS)

The link file for VI Windows products is provided in \xgf\src\xgfdos.run. This file must be invoked by a PostScript run command contained in the initialization file for PostScript interpreters running under a DOS file system.

The specified DOS drive can be changed to the drive on which VIPP® is loaded. In this case, the double backslashes (\\) are mandatory as the backslash is considered an escape character in a PostScript string.

```
%!PS-Adobe-2.0
%%Title: FreeFlow VI Compose link file for DOS (xgfdos.run)
%%Creator: JYB
%%CreationDate: Nov 2003
%%Copyright: (C) 1993,1994,1995,1996,1997,1998,1999,2000,2001,2002,2003,2004,
%%+ 2005,2006,2007,2008,2009,2011 Xerox Corporation. All Rights Reserved.
%%Source: $RCSfile: xgfdos.run,v $ $Revision: 1.47 $ - $Date: 2011/05/12 07:26:59 $
%%EndComments

(c:\\xgf\\src\\xgf.lic) run      % run xgf license
(c:\\xgf\\src\\xgf.eeh) run     % run xgf prolog
(c:\\xgf\\src\\vip.vsm) run     % run VI Compose Services Module

% NOTE:   The customer may add new paths or edit the original default paths
%         to reflect their local configuration. However, please DO NOT remove
%         any of the original default paths.

XGfdict begin                  % push XGfdict on stack

[ (c:\\xgfc\\formlib\\)        % customer path for forms & segments
  (c:\\xgf\\formlib\\)         % demo path for forms & segments
  (c:\\xgf\\gshared\\)         % demo path for globally shared forms & segments
] SETFPATH

[ (c:\\xgfc\\imglib\\)          % customer path for images
  (c:\\xgf\\imglib\\)          % demo path for images
  (c:\\xgfc\\decomp\\)         % path for decomp images
] SETIPATH

[ (c:\\xgfc\\jdtlib\\)          % customer path for jdt
  (c:\\xgf\\jdtlib\\)          % demo path for jdt
  (c:\\xgf\\gshared\\)         % demo path for globally shared jdt files
] SETJPATH

[ (c:\\xgfc\\encoding\\)        % customer path for encoding files
  (c:\\xgf\\encoding\\)        % demo path for encoding files
] SETEPATH

[ (c:\\xgfc\\mislib\\)          % customer path for miscell. files
  (c:\\xgfc\\fontlib\\)        % customer path to font lib
  (c:\\xgf\\demo\\)            % demo path for demo files
  (c:\\xgfc\\formlib\\)        % customer path for forms & segments
  (c:\\xgf\\formlib\\)        % demo path for forms & segments
  (c:\\xgf\\encoding\\)        % demo path for encoding files
  ()                          % null path for full names
] SETMPATH

% Note:   In SETPPATH, the path containing the directory component "vpcpfd" must
%         be the first path and no other path in SETPPATH can contain the "vpcpfd"
%         component.
```

```

[ (c:\\xgf\\vpcpfd\\$$FOLDER.\\$$PROJECT.\\) % project path for expanding print&forget VPCs
  (c:\\xgfc\\$$FOLDER.\\$$PROJECT.\\) % project local paths
  (c:\\xgfc\\$$FOLDER.\\fshared\\) % project folder shared paths
  (c:\\xgfc\\gshared\\) % project global shared paths
  (c:\\xgfc\\fontlib\\) % project access to font lib
  (c:\\xgf\\encoding\\) % project access to standard encoding
  (c:\\xgf\\gshared\\) % project global shared path
] SETPPATH

% ALERT! The xgfspool file MUST be writable.
% The following line may have to be edited to specify a path
% to the file xgfspool that is writable.

(c:\\xgfspool) SPOOLNAME % XGF spoolfile full name

% WARNING!!
% The following statements MUST NOT be moved.
% To do so may cause unpredictable results.
%
% They may be edited only to add or remove a leading comment notation
% (percent sign "%") to enable or disable the indicated features.

(c:\\xgf\\src\\xgf.gep) run % run xgf predef. color keys and GEP keys
(c:\\xgf\\src\\scoat.cck) run % run xgf solid coated color keys
(c:\\xgf\\src\\sucoat.cck) run % run xgf solid uncoated color keys
(c:\\xgf\\src\\xgf.ddg) run % run data driven graphics
(c:\\xgf\\src\\xgf.417) run % run PDF417 barcode
(c:\\xgf\\src\\xgf.max) run % run MaxiCode barcode
(c:\\xgf\\src\\xgf.dmx) run % run Datamatrix barcode
(c:\\xgf\\src\\xgf.qrc) run % run QR Code barcode
(c:\\xgf\\src\\xgf.4cb) run % run USPS4CB barcode
(c:\\xgf\\src\\cjk.def) run % run cjk default setting
(c:\\xgf\\src\\arb.def) run % run arb default setting
(c:\\xgf\\src\\xgf.def) run % run xgf default setting
(c:\\xgf\\src\\xgf.pcc) run % run xgf PCC definitions
(c:\\xgf\\src\\xgf.mup) run % run xgf multi-up definitions
(c:\\xgf\\src\\xgf.bat) run % run predefined BAT keys
(c:\\xgf\\src\\vippp.spi) run % run Specialty imaging resources
endrun % mandatory (remove XGFdict from dict stack)

```

## GENERAL DEFAULTS SETUP FILE

The general default setup file is provided in `/usr/xgf/src/xgf.def` on UNIX FFPS and `%XPS_HOME2%/resources/xgf/src/xgf.def` on Windows FFPS. This file is invoked from the link file and can be changed to accommodate specific site defaults and the local language.

```

%!PS-Adobe-2.0
%%Title: xgf.def
%%Creator: JYB/XECH
%%CreationDate: Apr 2000
%%For: VIPP default settings - rel. 4.1 (Letter)
%%Copyright: (C) 2000,2001,2002,2003,2004,2007,2009,2010,2011 Xerox Corp. All Rights Reserved.
%%Source: $RCSfile: xgf.def,v $ $Revision: 1.18 $ $Date: 2011/03/03 18:08:17 $
%%EndComments

```

```
%%BeginSetup
```

```

%-----
% Defaults setting
%-----

```

```

500                SETBUFSIZE          % line buffer length
(%%XGF)            SETNMP               % Native Mode Prefix (line mode)
(/)               SETFTSW              % font switch anchor
(:)               SETDBSEP             % set date base field delimiter
(fontlist) (win1252) SETENCODING        % set default re-encoded font list
(nullfl) null      SETENCODING          % non reencoded fontlist (Symbol,...)
DOT3 SETUNIT       % set unit to 1/300 inch
PORT              % set orientation to portrait
100 100 100 100    SETMARGIN            % set default margin
2350              SETCOLWIDTH          % set default column width
80 66             SETGRID              % set default grid
0 0 0             SETFRAME             % set no default frame
1 0 0             SETZEBRA             % set no default zebra
0 0 0             SETPAGENUMBER        % set no default page numbering
/PNFT /NHE 10     INDEXFONT            % set default page number font
/PNCL BLACK       INDEXCOLOR           % set default page number color
/NCR 0            SETFONT              % set auto size font mode with Courier
[ /DecimalPoint 46 % set period as decimal delimiter
  /FDecimalPoint 46 % set period as FORMAT decimal delimiter
  /FPunctuation 44 % set comma as FORMAT thousand delimiter
  /TimeZone -480    % set time zone, +- minutes from UTC. -480 = Pacific Standard
  /Daylightsaving [ % set start, end times for Daylight Saving Time
    [ 2003 60 96 120 299 120 ] % year adj. sday stime endday endtime
    [ 2004 60 95 120 305 120 ] % 2004 +1Hr day95 0200Hr day305 0200Hr (4 Apr - 26 Oct)
    [ 2005 60 93 120 303 120 ] % 2005 +1Hr day93 0200Hr day303 0200Hr (3 Apr - 30 Oct)
    [ 2006 60 92 120 302 120 ] % 2006 +1Hr day92 0200Hr day302 0200Hr (2 Apr - 29 Oct)
    [ 2007 60 70 120 308 120 ] % 2007 +1Hr day70 0200Hr day308 0200Hr (11 Mar - 4 Nov) 2nd
  Sunday
  [ 2008 60 69 120 307 120 ] % 2008 +1Hr day69 0200Hr day307 0200Hr (9 Mar - 2 Nov) in
  Mar
  [ 2009 60 67 120 305 120 ] % 2009 +1Hr day67 0200Hr day305 0200Hr (8 Mar - 1 Nov)
  thru
  [ 2010 60 73 120 311 120 ] % 2010 +1Hr day73 0200Hr day311 0200Hr (14 Mar - 7 Nov) 1st
  Sunday
  [ 2011 60 72 120 310 120 ] % 2011 +1Hr day72 0200Hr day310 0200Hr (13 Mar - 6 Nov)
  in Nov
  [ 2012 60 71 120 309 120 ] % 2012 +1Hr day71 0200Hr day309 0200Hr (11 Mar - 4 Nov)
  [ 2013 60 69 120 307 120 ] % 2013 +1Hr day69 0200Hr day307 0200Hr (10 Mar - 3 Nov)
  [ 2014 60 68 120 306 120 ] % 2014 +1Hr day68 0200Hr day306 0200Hr ( 9 Mar - 2 Nov)
  [ 2015 60 67 120 305 120 ] % 2015 +1Hr day67 0200Hr day305 0200Hr ( 8 Mar - 1 Nov)
  [ 2016 60 73 120 311 120 ] % 2016 +1Hr day73 0200Hr day311 0200Hr (13 Mar - 6 Nov)
  [ 2017 60 71 120 309 120 ] % 2017 +1Hr day71 0200Hr day309 0200Hr (12 Mar - 5 Nov)
  [ 2018 60 70 120 308 120 ] % 2018 +1Hr day70 0200Hr day308 0200Hr (11 Mar - 4 Nov)
  [ 2019 60 69 120 307 120 ] % 2019 +1Hr day69 0200Hr day307 0200Hr (10 Mar - 3 Nov)
  [ 2020 60 68 120 305 120 ] % 2020 +1Hr day68 0200Hr day305 0200Hr ( 8 Mar - 1 Nov)

```

```

/DayLong [(Sunday)(Monday)(Tuesday)(Wednesday)(Thursday)(Friday)(Saturday)]
/DayShort [(Sun)(Mon)(Tue)(Wed)(Thu)(Fri)(Sat)]
/MonthLong[(January)(February)(March)(April)(May)(June)(July)(August)(September)(October)
(November)(December)]
/MonthShort [(Jan)(Feb)(Mar)(Apr)(May)(Jun)(Jul)(Aug)(Sep)(Oct)(Nov)(Dec)]
/TimeZoneName[(PST)(PDT)] % Time zone names: Standard time, Daylight Saving time
/AmPm [(a.m.)(p.m.)]
/DefaultDate [ 2003 1 1 00 00 00 ] % 2003 Jan 1 00:00:00 (Standard time)
] SETPARAMS

GETDATE % capture initial date&time
%%EndSetup

```

## Utilities and Demonstration Files

Each VI Compose installation includes both utility and demonstration files. Demonstration files are either Golden Job or individual demonstration files. Descriptions of the utilities and file types, and a list of the demonstration files are found in these sections:

- [xgfpriint](#)
- [VI Project Container Filter](#)
- [VIPPManage](#)
- [VI Compose version information](#)
- [License Request forms](#)
- [VIPP demonstration files:](#)
  - [Golden Jobs](#)
  - [Individual demonstration files](#)
  - [Specialty Imaging applications](#)
  - [Third-Party emitters](#)
  - [Miscellaneous PostScript files](#)
- [Demonstration file list](#)

The utilities listed here are Bourne shell UNIX scripts that can run on most UNIX systems.

Enter the utility name without any parameters to display the usage statement and available options. Submit the PostScript and VIPP® demonstration files to any PostScript or VIPP® printer.

### XGFPRINT

The xgfpriint utility prints an ASCII file on a VIPP® printer using VIPP® capabilities. The syntax is as follows:

```
xgfpriint [ -jdt <jdtname> -cpl <cpl> -lpp <lpp> -tab <tablength> -auto -zebra -2up -dup -
tundup ]filename
```

#### Where:

jdtname	is the name of the JDT used to format the data. The default is pO.jdt, which is set up with these values: <ul style="list-style-type: none"> <li>• portrait 120, 120, 180, 150 (T, B, L, R) margins</li> <li>• cpl 85 (characters per line; default is 85)</li> <li>• lpp 70 (lines per page; default is 70)</li> <li>• Xerox logo top right</li> <li>• Courier font</li> </ul>
cpl	is the number of characters per line. The default is 85.
lpp	is the number of lines per page. The default is 70.
tablength	is the tabulator value expressed in number of characters. The default is 8.

filename	is the name of the file you want to print.
-auto	parses the print file to extract the cpl and lpp values. p0jdt is advised when using this option.
-zebra	selects Green Bar printing.
-2up	selects two-up mode printing.
-dup	selects duplex printing.
-tumdup	selects tumble-duplex printing.

## VI PROJECT CONTAINER FILTER

The VI Project Container Filter (VPCF) is a job filtering mechanism that can be applied to one or more queues on a FreeFlow Print Server system. When the VPC Filter is set up on a queue, it processes all VIPP® jobs sent to the queue via any print path gateway. Non-VPC files are passed through the queue unfiltered and VPC files are filtered / expanded, resources deployed, etc. according to the filter's setup configuration for the given queue. The VPCF GUI enables users to specify how the filter will process VPC files and manage their resources. Refer to [VI Project Container Filter and FreeFlow Print Server](#) for more information.



Note: FreeFlow Print Server also enables Watched Folders as a job submission method. You can configure a Watched folder to point to a queue that has been enabled with a VPCF filter option to process .vpc files.

## VIPP® MANAGE

VIPP® Manage is a Windows based utility that provides a GUI for the management of VIPP® resources on the disk of a VIPP®-enabled desktop or office print device, and for managing VIPP® jobs and resources on diskless printers.

## VI COMPOSE VERSION INFORMATION

VIIVersionInfo.nm is a utility used to print version number information for VI Compose and its components. This information is requested when submitting a SPAR (Software Problem Action Request) to engineering.

To submit a SPAR please contact the local Xerox analyst or hotline to open a log. Supply the version information reported by the VIIVersionInfo.nm utility as part of the SPAR information.

## LICENSE REQUEST FORMS

The License Request form, either vipplrfus.ps (US) or vipplrfxe.ps (Europe), is located in the top VIPP® directory. To order VI Compose for a specific site, print this file on each printer and device on which VIPP® will run. Complete the required information on the printed forms. Use this form as instructed by a Xerox representative

## VIPP® DEMONSTRATION FILES

Use VIPP® demonstration files for demonstrations and training exercises. These files are located in the VIPP® demonstration directory `/usr/xgf/demo` and in VIPP® application directories, such as `C:\vide\xgf/demo`. Required resources are located in the appropriate demonstration libraries such as `/usr/xgf/formlib`. Other

files used for demonstration are also located in the demo directory; those files include VIPP® Project Container (.vpc) files, and PDF files containing the output of the native, line or database mode demonstration files.

### Golden Jobs

Golden Jobs are delivered with VI Compose, they are used to send test jobs to your printer.

All golden jobs except `goljobv` require that these media types be assigned to specific trays on the target printer using the appropriate mechanism for that printer:

- YELLOW
- GREEN
- Default paper

Use `goljobv` or `gjobcv.nm` to verify the VI Compose installation without loaded media.

### Individual demonstration files

Individual demonstration files are part of the golden jobs and can also be run individually. They include line, native, and database mode files.

### Specialty Imaging applications

Specialty Imaging application files are provided in support of the Specialty Imaging feature.

### Third-Party emitters

Third party VIPP® emitters are design tools or document applications that emit or use VIPP®. VIPP® Designers make VIPP® job design easier, using drag and drop GUIs to design the VIPP® templates and job resources.

Many third-party emitters also enable remote resource management on office printers. To run VIPP® on an office print device review the tools that support loading VIPP® to those devices.

Contact a Xerox representative for more information on supported VI tools.

### Miscellaneous PostScript files

Miscellaneous PostScript files are also listed in the table below. For further information, refer to [Standard lists, tables, keys, and attributes](#).

## DEMONSTRATION FILE LIST

The table below contains a list of the files found in the `./xg£/demo` directory, and a brief description of each.

FILE NAME	DESCRIPTION	GOL-D JOB	MISCEL- LANE- OUS	SPECIALTY IMAGING	THIRD- PARTY EMITTER
billb.lm	Invoices using RPE prefixed data processing options (blue)		x		
billg.lm	Invoices using RPE prefixed data processing options (green)		x		
billr.lm	Invoices using RPE prefixed data processing options (red)		x		
cutmark.nm	Demonstrate usages of the CUTMARK command		x		

FILE NAME	DESCRIPTION	GOLD JOB	MISCELLANEOUS	SPECIALTY IMAGING	THIRD-PARTY EMITTER
dynbox.nm	Demonstrate how to make a dynamic box using SAVEPP and SVPOS		x		
dynbox3.lm	Demonstrate how to make dynamic boxes in line mode		x		
gjobcv.nm	Black native mode golden job	x			
gjobcv6135.nm	FreeFlow Print Server native mode golden job	x			
gjobcvb.nm	Blue native mode golden job	x			
gjobcvg.nm	Green native mode golden job	x			
gjobcvr.nm	Red native mode golden job	x			
goljob	Black golden job	x			
goljob6135	FreeFlow Print Server golden job	x			
goljobb	Blue golden job	x			
goljobg	Green golden job	x			
goljobr	Red golden job	x			
goljobv	Verify golden job, a black golden job without tray requirements	x			
imagedemoc.nm	Bank statements with images using native mode		x		
imgdemo.nm	Bank statements with images using native mode	x			
LayoutSimulator.nm	A layout definition simulator, prints page layouts				
letter.dbf	A mailing using database mode	x			
lis0.lm	Sales reports using listing data processing options		x		
lis1.lm	Sales reports using listing data processing options		x		
lis1b.lm	Sales reports using listing data processing options		x		
lis2.lm	Sales reports using listing data processing options		x		

FILE NAME	DESCRIPTION	GOLD JOB	MISCELLANEOUS	SPECIALTY IMAGING	THIRD-PARTY EMITTER
mulbeg.lm	JDT sample to simulate multiple begins feature		x		
number_2_words.dbf	Sample of VIPP® code to convert a number into words		x		
palrgb.ps	Color samples with RGB coding references. Used to define new Colorkeys.				x
palrgb2.ps	Color samples with RGB coding references. Used to define new Colorkeys.				x
palrgb3.ps	Color samples with RGB coding references. Used to define new Colorkeys.				x
rpe2.lm	Sales report using RPE data processing options that includes nested conditions and variable line spacing	x			
rpe2b.lm	Sales reports using RPE data processing options and cyclecopy		x		
rpe3.lm	Sales reports using RPE data processing options and cyclecopy		x		
rpe4.lm	Sales reports using RPE data processing options and cyclecopy		x		
rpe5.lm	Sales reports using RPE data processing options and cyclecopy	x			
rpe5b.lm	Sales reports using RPE data processing options and cyclecopy		x		
sam417.nm	PDF417 barcode examples		x		
samaztec.nm	Aztec barcode examples				
sambat_v.ps	Vertical BATKey samples				x
sambat.ps	Predefined gray Background Attributes BATkey samples.				x
sambatb.ps	Predefined blue BATkey samples.				x
sambatg.ps	Predefined green BATkey samples.				x
sambatr.ps	Predefined red BATkey samples.				x
samcc.nm	VIPP® pattern samples		x		
samccc.ps	VIPP® Solid Coated Color simulation chart		x		

FILE NAME	DESCRIPTION	GOLD JOB	MISCELLANEOUS	SPECIALTY IMAGING	THIRD-PARTY EMITTER
	with CMYK values				
samccu.ps	VIPP® Solid Uncoated Color simulation chart with CMYK values		x		
samdatabar.nm	Databar barcode examples		x		
samddg.lm	Palette of Data Driven Graphics (DDG) samples bank statements that demonstrate the usage of dynamic boxes and Data Driven Graphics		x		
samddg.nm	Palette of Data Driven Graphics (DDG) sample business charts		x		
samddg.pdf	Contains the output of samddg.nm		x		
samddg.ps	Palette of Data Driven Graphics (DDG) samples				x
samddgs.lm	Bank statement that demonstrates the usage of dynamic boxes and Data Driven Graphics	x			
samdmx.nm	DataMatrix barcode examples		x		
samfont.ps	Character set sample for each of the fonts listed in the files xgf/encoding/fontlist and xgf/encoding/nullfl				x
samgep.ps	Predefined gray GEPkey samples. Refer to <a href="#">Standard lists, tables, keys, and attributes</a> for further information.				x
samgepb.ps	Predefined blue GEPkey samples.				x
samgepg.ps	Predefined green GEPkey samples.				x
samgepr.ps	Predefined red GEPkey samples.				x
samlkf.nm	Sample of SETLKF usage.				x
sammax.nm	Maxicode barcode examples		x		
samomr.nm	Test of OMR code for mailing machine		x		
sampat.nm	Palette of predefined pattern samples.				x
sampff.vpc	Demonstrate usages of the DRAWPFF command		x		
sampif.nm	VIPP® Interactive PDF samples		x		

FILE NAME	DESCRIPTION	GOLD JOB	MISCELLANEOUS	SPECIALTY IMAGING	THIRD-PARTY EMITTER
sampif.pdf	Output from sampif.nm in PDF format		x		
samqrc.nm	QR code examples		x		
samxml1.vpc	VI Project container (VPC) containing an example of an XML job		x		
samxml2.vpc	VI Project container (VPC) containing an example of an XML job.				x
SI_CorrColor-PatchBook.ps	87 page color patch book for Correlation. Prints color palettes used in Specialty imaging. Refer to <a href="#">Specialty Imaging with VIPP</a> for further information.			x	x
SI_CR_2L_Samples.nm	Sample Two Layer Correlation job			x	
SI_GlossColor-PatchBook.ps	87 page color patch book for GlossMark Text. Prints color palettes used in Specialty imaging. Refer to <a href="#">Specialty Imaging with VIPP</a> for further information.			x	x
SI_IR_2L_Samples.nm	Sample Two Layer Infrared Job			x	
SI_IR_Samples.nm	Sample Infrared Job			x	
SI_UV_Samples.nm	Sample Fluorescent Job			x	
SI_text_patterns.vpc	SETTPAT and UV2L test case		x		
SI_VP_GlossMark.vpc	Samples of text and colors used in GlossMark color applications.			x	
SI_VP_Correlation_1L.vpc	Samples of text and colors used in single layer Correlation Mark applications.			x	
SI_VP_Correlation_2L.vpc	Samples of text and colors used in two-layer Correlation Mark applications.			x	
SI_VP_Correlation_key.nm	Sample of vector pattern correlation key.			x	

FILE NAME	DESCRIPTION	GOLD JOB	MISCELLANEOUS	SPECIALTY IMAGING	THIRD-PARTY EMITTER
SI_VP_VoidPanto.vpc	Sample of void pantograph vector pattern.			x	
subset.dbf	Subset finishing test file	x			
subset1.lm	Subset finishing test file	x			
subset2.lm	Subset finishing test file	x			
tables.lm	Mix of pc8 and pcsun encoding tables		x		
truk.nm	Prints a TIFF image		x		
USPS4CBtest.nm	USPS4CB barcode test		x		
VIIVersionInfo.nm	Prints VI Compose component version numbers		x		
xgf2.ps	Microsoft PowerPoint compatibility example		x		
xgf3.nm	Microsoft PowerPoint compatibility example		x		
ZSORT_Example.vpc	VI Project Container with a ZSORT example		x		

# VIPP® File Examples

This chapter contains:

Example 1: Bank Statement with Images Using Native Mode .....	134
Example 2: Sales Report Using Line Mode.....	137
Example 3: Sales Report Using RPE Line Mode.....	140
Example 4: Invoice Using RPE Prefixed Line Mode.....	144
Example 5: Mailing Using Database Mode.....	151
Example 6: Bank Statement Using Dynamic Boxes and Data-Driven Graphics .....	154

This chapter contains these VIPP® file examples:

- Example 1: Bank statement with images using native mode
- Example 2: Sales report using line mode
- Example 3: Sales report using RPE line mode
- Example 4: Invoice using RPE prefixed line mode
- Example 5: Mailing using database mode
- Example 6: Bank statement using dynamic boxes and Data Driven Graphics

The electronic files are located in the corresponding VIPP® directories.

## Example 1: Bank Statement with Images Using Native Mode

This section contains an example of a bank statement with images using native mode. The example consists of a print file, form file, and an example of the generated output.

### PRINT FILE EXAMPLE

```
%!
PORT
(example1.frm) SETFORM
/NHE 12 SETFONT
50 SETLSP
500 2820 MOVETO (1284500600) SH
1500 3000 MOVETO
(Hr. Kurt Miller) SHL
(6 rue des Bois) SHL
(1227 Carouge) SHL
1900 2355 MOVETO (1) SHR
1900 2255 MOVETO (6200.00) SHR
1900 2155 MOVETO (6200.00) SHR
1900 2055 MOVETO (0.00) SHR
1900 1555 MOVETO (2) SHR
1900 1455 MOVETO (670.00) SHR
1900 1355 MOVETO (660.00) SHR
1900 1255 MOVETO (10.00) SHR
1900 755 MOVETO (3) SHR
1900 655 MOVETO (1423.60) SHR
1900 555 MOVETO (1423.60) SHR
1900 455 MOVETO (0.00) SHR
200 2600 MOVETO
(bvr3.tif) 1 0 ICALL
PAGEBRK
```

## FORM FILE EXAMPLE

```

%%title: example1.frm
{
PORT
200 3200 MOVETO
(trdbank.tif) 1 0 ICALL
200 3150 2150 0 S1 DRAWB
197 2903 856 126 S1 DRAWB
197 3053 856 126 LT_S1 DRAWB
1447 1000 906 750 S1 3 0 800 DRAWBM
1450 350 900 50 LT 7 0 100 DRAWBM
1450 1150 900 50 LT 7 0 100 DRAWBM
1450 1950 900 50 LT 7 0 100 DRAWBM
/NHE 9 SETFONT
1460 2560 MOVETO (wir haben Ihrem Konto den nebenan) SH
1460 2510 MOVETO (abgebildeten Posteingang wie folgt) SH
1460 2460 MOVETO (gutgeschrieben:) SH
1460 2360 MOVETO (Beleg-Nr. :) SH
1460 2260 MOVETO (Betrag:) SH
1460 2160 MOVETO (Valuta:) SH
1460 2060 MOVETO (Postspesen:) SH
1460 1760 MOVETO (wir haben Ihrem Konto den nebenan) SH
1460 1710 MOVETO (abgebildeten Posteingang wie folgt) SH
1460 1660 MOVETO (gutgeschrieben:) SH
1460 1560 MOVETO (Beleg-Nr. :) SH
1460 1460 MOVETO (Betrag:) SH
1460 1360 MOVETO (Valuta:) SH
1460 1260 MOVETO (Postspesen:) SH
1460 960 MOVETO (wir haben Ihrem Konto den nebenan) SH
1460 910 MOVETO (abgebildeten Posteingang wie folgt) SH
1460 860 MOVETO (gutgeschrieben:) SH
1460 760 MOVETO (Beleg-Nr. :) SH
1460 660 MOVETO (Betrag:) SH
1460 560 MOVETO (Valuta:) SH
1460 460 MOVETO (Postspesen:) SH
/NHEB 9 SETFONT
220 2970 MOVETO (GUTSCHRIFTSANZEIGE) SH
220 2820 MOVETO (KONTO NR.) SH
/NHE 9 SETFONT
220 2660 MOVETO (GESCHÄFTSKONTO SFR) SH
/NHE 6 SETFONT
190 300 MOVETO (Printed on Xerox DocuPrint) 90 FBLACK 0 SHX
} FSHOW

```



Note: The **FSHOW** command caches the form file automatically. When the form file contains a variable reference, delete the **FSHOW** command. When **FSHOW** is not deleted, only the first instance of the variable object is cached and used for subsequent calls.

This figure illustrates the printed output for Example 1.

# TradeBank Inc.

GUTSCHRIFTSANZEIGE

KONTO NR. 1284500600

Hr. Kurt Müller  
6 rue des Bois  
1227 Carouge

GESCHÄFTSKONTO SFR

Facture 6000

22-436

6700

ATP SA  
r. des Gares  
1227 Genève

Wir haben Ihrem Konto den nebenan  
abgebildeten Posteingang wie folgt  
gutgeschrieben:

Beleg-Nr.: 1

Betrag: 6200.00

Valuta: 6200.00

Postspesen: 0.00

Facture 2207

22-27

670

M. Jacques Brutti  
r. du Temple 6  
1227 Carouge

Wir haben Ihrem Konto den nebenan  
abgebildeten Posteingang wie folgt  
gutgeschrieben:

Beleg-Nr.: 2

Betrag: 670.00

Valuta: 660.00

Postspesen: 10.00

Facture 2207

22-47

7423 60

Mme. A. Brutti  
r. du Temple 4  
1227 Carouge

Wir haben Ihrem Konto den nebenan  
abgebildeten Posteingang wie folgt  
gutgeschrieben:

Beleg-Nr.: 3

Betrag: 1423.60

Valuta: 1423.60

Postspesen: 0.00

## Example 2: Sales Report Using Line Mode

This section contains an example of a sales report using basic line mode. This example consists of a print file, a JDT file, and an example of the generated output.

### PRINT FILE EXAMPLE

```
%!
(example2.jdt) STARTLM
SALES REPORT - Europe
Product Division
France
United Kingdom
Germany
Spain
Italy
Belgium
Total

BUSINESS DIVISIONS

Personal Document Products 230.120 340.200 190.200 210.000 103.000 194.000 2.234.000
Office Document Systems 210.000 540.000 200.400 123.040 95.050 23.000 1.200.000
Office Document Products 210.000 540.000 200.400 123.040 95.050 23.000 1.200.000
Xsoft 230.120 340.200 190.200 210.000 103.000 94.000 2.234.000
Document Production Systems 210.000 540.000 200.400 123.040 95.050 23.000 1.200.000
Document Printing Systems 210.000 540.000 200.400 123.040 95.050 23.000 1.200.000
Xerox Engineering Systems 230.120 340.200 190.200 210.000 103.000 94.000 2.234.000
Xerox Business Services 210.000 540.000 200.400 123.040 95.050 23.000 1.200.000
Office Document Services 30.120 340.200 190.200 210.000 103.000 94.000 2.234.000

CUSTOMER OPERATION DIVISIONS

U.S. Customer Operations 230.120 340.200 190.200 210.000 103.000 94.000 2.234.000
Americas Customer Operations 210.000 540.000 200.400 123.040 95.050 23.000 1.200.000
U.S. Customer Operations 230.120 340.200 190.200 210.000 103.000 94.000 2.234.000
TOTAL 1.230.120 6.340.200 3.190.200 1.210.000 1.103.000 594.000 19.234.000
%%EOF
```

## JOB DESCRIPTOR TICKET FILE EXAMPLE

```

%!PS-Adobe-2.0
%%Title: example2.jdt
%%Creator: JYB/RXCH
%%CreationDate: MAR 97
%%For: JDT definition for Variable Data Intelligent PostScript Printware
%%Copyright: (C) 1995 by Rank Xerox AG (RXCH). All right reserved.
%%EndComments

%-----
% set orientation
%-----
ILAND                      % PORT - Portrait    IPORT - Inverse Portrait
                           % LAND - Landscape    ILAND - Inverse Landscape

%-----
% set margins
%-----
200                        % top margin
200                        % bottom margin
200                        % left margin
200                        % right margin
SETMARGIN

%-----
% set grid
%-----
103                        % number of characters per line
33                         % number of line per page
SETGRID

%-----
% set frame
%-----
3                          % frame width (0 = no frame)
50                         % frame offset from margins
BLACK                     % frame color (Colorkey)
30                        % corner radius ( 0 = square corners)
SETFRAME

%-----
% set zebra and form
%-----
LIGHT                     % zebra color (Colorkey, WHITE = no zebra)
3                         % number of lines with zebra
3                         % number of lines without zebra
SETZEBRA
{ ILAND 3110 2273 MOVETO (xdlogo.tif) .66 0 ICALL } SETFORM

%-----
% set page numbering
%-----
/PNFT /NHE 10 INDEXFONT   % PN font (PNFT is a reserved font index)
/PNCL BLACK INDEXCOLOR    % PN color (PNCL is a reserved color index)
(- ## -)                 % print format (# stands for page number)
1                         % starting page number (<1 will not print)
1                         % 0 - do not print page number
                           % 1 - bottom center      2 - bottom right
                           % 3 - top center        3 - top right
SETPAGENUMBER

%-----
% set font
%-----
/NCR                      % font name (key)
0                          % font size in points (1/72 inch)
                           % 0 = font size will be computed from SETGRID
SETFONT

```

The following figure illustrates the printed output for Example 2.

SALES REPORT - Europe							
Product Division							
France							
Uted Kingdom							
Germany							
Spain							
Italy							
Belgium							
Total							
BUSINESS DIVISIONS							
Personal Document Products	230.120	340.200	190.200	210.000	103.000	194.000	2.234.000
Office Document Systems	210.000	540.000	200.400	123.040	95.050	23.000	1.200.000
Office Document Products	210.000	540.000	200.400	123.040	95.050	23.000	1.200.000
Xsoft	230.120	340.200	190.200	210.000	103.000	94.000	2.234.000
Document Production Systems	210.000	540.000	200.400	123.040	95.050	23.000	1.200.000
Document Printing Systems	210.000	540.000	200.400	123.040	95.050	23.000	1.200.000
Xerox Engineering Systems	230.120	340.200	190.200	210.000	103.000	94.000	2.234.000
Xerox Business Services	210.000	540.000	200.400	123.040	95.050	23.000	1.200.000
Advanced Office Document Services	30.120	340.200	190.200	210.000	103.000	94.000	2.234.000
CUSTOMER OPERATION DIVISIONS							
U.S. Customer Operations	230.120	340.200	190.200	210.000	103.000	94.000	2.234.000
Americas Customer Operations	210.000	540.000	200.400	123.040	95.050	23.000	1.200.000
U.S. Customer Operations	230.120	340.200	190.200	210.000	103.000	94.000	2.234.000
TOTAL	5.230.120	6.340.200	3.190.200	1.210.000	1.103.000	594.000	19.234.000

### Example 3: Sales Report Using RPE Line Mode

This section contains an example of a sales report using RPE line mode. The example consists of a print file, a form file, a JDT, and an example of the generated output.

#### PRINT FILE EXAMPLE

```

%!
(example3.jdt) STARTLM
SALES REPORT - Europe
Product Division
France
United Kingdom
Germany
Spain
Italy
Belgium
Total

BUSINESS DIVISIONS

Personal Document Products 230.120 340.200 190.200 210.000 103.000 194.000 2.234.000
Office Document Systems 210.000 540.000 200.400 123.040 95.050 23.000 1.200.000
Office Document Products 210.000 540.000 200.400 123.040 95.050 23.000 1.200.000
Xsoft 230.120 340.200 190.200 210.000 103.000 94.000 2.234.000
Document Production Systems 210.000 540.000 200.400 123.040 95.050 23.000 1.200.000
Document Printing Systems 210.000 540.000 200.400 123.040 95.050 23.000 1.200.000
Xerox Engineering Systems 230.120 340.200 190.200 210.000 103.000 94.000 2.234.000x
Xerox Business Services 210.000 540.000 200.400 123.040 95.050 23.000 1.200.000
Office Document Services 30.120 340.200 190.200 210.000 103.000 94.000 2.234.000
CUSTOMER OPERATION DIVISIONS

U.S. Customer Operations 230.120 340.200 190.200 210.000 103.000 94.000 2.234.000
Americas Customer Operations 210.000 540.000 200.400 123.040 95.050 23.000 1.200.000
U.S. Customer Operations 230.120 340.200 190.200 210.000 103.000 94.000 2.234.
000TOTAL 5.230.120 6.340.20 3.190.200 1.210.000 1.103.000 594.000 19.23
4.000
%%EOF

```

## FORM FILE EXAMPLE

```

%%title: example3.frm
{ ILAND
3309 2293 MOVETO (xdlogo.tif) .7 0 21 ICALL
 210 330 825 150 LT 6 0 300 DRAWBM
1047 330 313 150 LT 6 0 300 DRAWBM
1372 330 300 150 LT 6 0 300 DRAWBM
1685 330 300 150 LT 6 0 300 DRAWBM
 210 2105 825 100 FBLACK DRAWB
1047 2105 313 100 FBLACK DRAWB
1372 2105 300 100 FBLACK DRAWB
1685 2105 300 100 FBLACK DRAWB
1997 2105 300 100 FBLACK DRAWB
2310 2105 313 100 FBLACK DRAWB
2635 2105 325 100 FBLACK DRAWB
2972 2105 337 100 FBLACK DRAWB
 210 1980 825 1875 S1 DRAWB
1047 1980 313 1875 S1 DRAWB
1372 1980 300 1875 S1 DRAWB
1685 1980 300 1875 S1 DRAWB
1997 1980 300 1875 MED_S1 DRAWB
2310 1980 312 1875 MED_S1 DRAWB
2635 1980 325 1875 MED_S1 DRAWB
2972 1980 337 1875 MED_S1 DRAWB
 210 2259 1262 100 S2 DRAWB
} FSHOW

```



Note: The **FSHOW** command caches the form file automatically. When the form file contains a variable reference, delete the **FSHOW** command. When **FSHOW** is not deleted, only the first instance of the variable object is cached and used for subsequent calls.

## JOB DESCRIPTOR TICKET FILE EXAMPLE

```

%!PS-Adobe-2.0
%%Title: example3.jdt
%%Creator: JYB/RXCH
%%CreationDate: MAR 97
%%For: RPE definition for Variable Data Intelligent PostScript Printware
%%Copyright: (C) 1995 by Rank Xerox AG (RXCH). All right reserved.
%%EndComments

%-----
% set orientation and margins
%-----
ILAND          % PORT - Portrait   IPORT - Inverse Portrait
               % LAND - Landscape  ILAND - Inverse Landscape
0 0 0 0 SETMARGIN  % set all margins to zero

%-----
% set form
%-----
(example3.frm) SETFORM

%-----
% set RPE fonts
%-----
/F1 /NHEB  9 13 INDEXFONT
/F2 /NHEB 11 15 INDEXFONT
/F2t /NHEB 20 15 INDEXFONT
/F3 /NHEB 13 15 INDEXFONT
/F4 /NHEB 18   INDEXFONT
/F5 /NHEB 13   INDEXFONT

%-----
% Record Criteria Definitions
%-----
% Cond. key   Rec.pos.   Length   cond.   ref. string
% /CND1       0         5         /eq     (TOTAL)   SETRCD

% RPE definition
%-----

5 BEGINRPE

% Almt rot.  xinit  xdispl  yinit  ydisp  pos.  length  Font  Color
1 FROMLINE
[ 2 0      835    0      300    0      00      99    /F4  BLACK ]
2 FROMLINE
[ 2 0      615    0      445    0      00      99    /F1  WHITE ]
3 FROMLINE
[ 2 0     1199   318     445    0      00      99    /F1  WHITE ]
9 FROMLINE
[ 2 0     3140    0      445    0      00      99    /F5  WHITE ]

10 FROMLINE
/CND1 [ 0 0      230    0      560    75      00      28    /F2t  BLACK ]
/ELSE [ 0 0      230    0      560    75      00      28    /F2   BLACK ]
[ 1 0      1345    0      560    75      28      10    /F2   BLACK ]
[ 1 0      1658    0      560    75      38      11    /F2   BLACK ]
[ 1 0      1976    0      560    75      49      11    /F2   BLACK ]
[ 1 0      2286    0      560    75      60      11    /F2   BLACK ]
[ 1 0      2610    0      560    75      71      11    /F2   BLACK ]
[ 1 0      2945    0      560    75      82      10    /F2   BLACK ]
[ 1 0      3290    0      560    75      92      11    /F3   BLACK ]

ENDRPE

```

The following figure illustrates the printed output for Example 3.

**xerox**

<b>SALES REPORT - Europe</b>							
Product Division	France	United Kingdom	Germany	Spain	Italy	Belgium	Total
<b>BUSINESS DIVISIONS</b>							
Personal Document Products	230.120	340.200	190.200	210.000	103.000	194.000	2.234.000
Office Document Systems	210.000	540.000	200.400	123.040	95.050	23.000	1.200.000
Office Document Products	210.000	540.000	200.400	123.040	95.050	23.000	1.200.000
Xsoft	230.120	340.200	190.200	210.000	103.000	94.000	2.234.000
Document Production Systems	210.000	540.000	200.400	123.040	95.050	23.000	1.200.000
Document Printing Systems	210.000	540.000	200.400	123.040	95.050	23.000	1.200.000
Xerox Engineering Systems	230.120	340.200	190.200	210.000	103.000	94.000	2.234.000
Xerox Business Services	210.000	540.000	200.400	123.040	95.050	23.000	1.200.000
Advanced Office Document Services	30.120	340.200	190.200	210.000	103.000	94.000	2.234.000
<b>CUSTOMER OPERATION DIVISIONS</b>							
U.S. Customer Operations	230.120	340.200	190.200	210.000	103.000	94.000	2.234.000
Americas Customer Operations	210.000	540.000	200.400	123.040	95.050	23.000	1.200.000
U.S. Customer Operations	230.120	340.200	190.200	210.000	103.000	94.000	2.234.000
<b>TOTAL</b>	<b>5.230.120</b>	<b>6.340.200</b>	<b>3.190.200</b>	<b>1.210.000</b>	<b>1.103.000</b>	<b>594.000</b>	<b>19.234.000</b>

## Example 4: Invoice Using RPE Prefixed Line Mode

The following is an example of an invoice using RPE prefixed line mode. The example consists of a print file, a form file, a JDT, and an example of the generated output.

### PRINT FILE EXAMPLE

```
%!
(example4.jdt) STARTLM
PER0 01.11.1989 - 31.12.1989
REF0 14153 01764960

ADR0 M. MARCEL DUPONT
ADR0 RESIDENCE "LES MIMOSA"
ADR0 PLACE DE LA GARE, 44
ADR0 2323 SAGEX

DFA0 23.01.1990
DPY0 23.02.1990
DRE0 7.12.1989
CRN0 068.025.000 COMMUNS IMMEUBLE

LFA0 14 COMMUNS D'IMMEUBLES      81978   30248   30144       104           4,40
LFA0 41 EAU TARIF I                491    12869   12714       155    0,2800    43,40
LFA0   TAXE DE BASE                 2,0     7,692     15,35
LFA0   LOCATION COMPTEUR            2,0     2,00     4,00
LFA0 82 GAZ CHAUFFAGE              3500   40972   40126   10,40    8798    0,0330    290,35
LFA0   TAXE DE BASE                 2,0    16,666     33,35
LFA0
LFA1 TOTAL S.I.                                430,85
LFA0
LFA0 41 ASSAINISSEMENT              491                                155     0,25     38,75

MNT0 MONTANT A PAYER          469,60

BRE0 00 00014 15301 76496 00120 01019

OLI0 0100000469609>000001415301764960012001019+
%%EOF
```

## FORM FILE EXAMPLE

%%Title: example4.frm

{ PORT

```

  0 1250 710 1250 XLT DRAWB
710 1250 1770 950 XLT DRAWB
710 1250 0 1250 D1 DRAWB
  0 1250 2480 0 D1 DRAWB
  0 1200 2480 0 S1 DRAWB
1250 1200 0 900 S1 DRAWB
2080 1200 0 300 S1 DRAWB
1250 900 1230 0 S1 DRAWB
2165 1160 220 220 D1 110 DRAWBR
 110 300 220 220 D1 110 DRAWBR

```

/NHEB 8 SETFONT

355 1210 MOVETO (Empfangsschein/Récépissé/Ricevuta) SHC

900 MOVEH (Einzahlung Giro PTT) SH

1350 MOVEH (Versement Virement PTT) SH

1900 MOVEH (Versamento Girata PTT) SH

/NHE 6 SETFONT

25 SETLSP

20 1160 MOVETO

(Einzahlung für / Versement pour / Versamento per) SH

750 1160 MOVETO

(Einzahlung für / Versement pour /) SHL

(Versamento per) SH

1270 1160 MOVETO

(Bitte keine Mitteilungen anbringen) SHL

(Pas de communications s.v.p.) SHL

(Non aggiungete comunicazioni p.f.) SH

20 890 MOVETO (Konto) SHL (compte) SHL (conto) SHL

750 890 MOVETO (Konto) SHL (compte) SHL (conto) SHL

1270 870 MOVETO (einbezahlt von) SHL (versé par) SHL (versato da) SH

1480 870 MOVETO (oder) SHL (ou) SHL (o) SH

1560 870 MOVETO (Giro aus Konto) SHL (virement du compte) SHL (girata dal conto) SH

/NHEB 8 SETFONT

1835 MOVEH (N°) SH

1870 815 400 0 D1 DRAWB

20 770 350 80 S3 2 720 0 DRAWBM

385 770 115 80 S3 2 720 0 DRAWBM

20 780 MOVETO (Fr.) SH 385 MOVEH (c.) SH

740 MOVEH (Fr.) SH 1105 MOVEH (c.) SH

/NHE 6 SETFONT

20 660 MOVETO (Einbezahlt von / Versé par / Versato da) SH

400 250 MOVETO

(Die Annahmestelle) SHL

(L'office de dépôt) SHL

(L'ufficio d'accettazione) SH

1270 780 0 90 S1 2 1050 0 DRAWBM

1270 690 1050 0 S1 2 0 70 DRAWBM

1270 780 270 0 S1 2 780 0 DRAWBM

```

/NHEC 6 SETFONT
1795 775 MOVETO (Referenz-Nr / N° de référence / N° di riferimento) SHC
POINT SETUNIT
% logo SIG
37 821 7 27 FBLACK 2 DRAWBR
42 821 18 8 FBLACK 2 DRAWBR
72 821 8 8 FBLACK 2 DRAWBR
82 821 8 36 FBLACK 2 DRAWBR
88 821 18 8 FBLACK 2 DRAWBR
46 811 24 7 FBLACK 2 DRAWBR
72 811 8 26 FBLACK 2 DRAWBR
92 807 23 7 FBLACK 2 DRAWBR
62 806 8 21 FBLACK 2 DRAWBR
108 804 7 19 FBLACK 2 DRAWBR
42 802 18 8 FBLACK 2 DRAWBR
37 792 28 7 FBLACK 2 DRAWBR
87 792 24 7 FBLACK 2 DRAWBR
% zebra -----
12 366 167 11 LT 11 0 24 DRAWBM
183 366 45 11 LT 11 0 24 DRAWBM
232 366 58 11 LT 11 0 24 DRAWBM
295 366 52 11 LT 11 0 24 DRAWBM
351 366 40 11 LT 11 0 24 DRAWBM
395 366 60 11 LT 11 0 24 DRAWBM
459 366 47 11 LT 11 0 24 DRAWBM
510 366 74 11 LT 11 0 24 DRAWBM
% Box total -----
510 332 74 25 LT_S1 3 DRAWBR
% Box reference -----
36 757 115 24 LT_S1 3 DRAWBR
% Box headers -----
12 649 167 19 FBLACK 3 DRAWBR
183 649 45 19 FBLACK 3 DRAWBR
232 649 58 19 FBLACK 3 DRAWBR
295 649 52 19 FBLACK 3 DRAWBR
351 649 40 19 FBLACK 3 DRAWBR
395 649 60 19 FBLACK 3 DRAWBR
459 649 47 19 FBLACK 3 DRAWBR
510 649 74 19 FBLACK 3 DRAWBR
% Box body -----
12 626 167 289 S1 3 DRAWBR
183 626 45 289 S1 3 DRAWBR
232 626 58 289 S1 3 DRAWBR
295 626 52 289 S1 3 DRAWBR
351 626 40 289 S1 3 DRAWBR
395 626 60 289 S1 3 DRAWBR
459 626 47 289 S1 3 DRAWBR
510 626 74 289 S1 3 DRAWBR
WHITE SETTXC
/NHEB 14 10 SETFONT
95.5 636 MOVETO (Libellé) SHC
/NHEB 12 10 SETFONT
547 636 MOVETO (Montant) SHC
/NHEB 7 10 SETFONT
261 636 MOVETO (Nouvel index) SHC
371 636 MOVETO (Coeffic.) SHC
321 636 MOVETO (Ancien index) SHC
/NHEB 6 7 SETFONT
205.5 641 MOVETO (No) SHC
425 641 MOVETO (Quantité) SHC
482.5 641 MOVETO (Prix unitaire) SHC
205.5 634 MOVETO (compteur SI) SHC
425 634 MOVETO (Consommée) SHC
482.5 634 MOVETO (Fr. Ct.) SHC
BLACK SETTXC

```

```

/NHE 09 SETFONT
345.6 789 MOVETO (Période:) SH
137 785.3 MOVETO (Tél. 093/221.56.77) SH
61.2 764 MOVETO (No de référence) SH
35 717 MOVETO (Date de facture:) SH35 705 MOVETO (Payable jusqu'au:) SH
35 686 MOVETO (Relevé du:) SH
35 662 MOVETO (Concerne:) SH
11 256 MOVETO (Services Industriels ) SH
183 256 MOVETO (Services Industriels ) SH
11 246 MOVETO (de la ville de Sagex) SH
183 246 MOVETO (de la ville de Sagex) SH
11 236 MOVETO (2323 Sagex) SH
183 236 MOVETO (2323 Sagex) SH
/NHE 13 12 SETFONT
137 813 MOVETO (SERVICES INDUSTRIELS) SH
137 799 MOVETO (2323 SAGEX) SH
/NHEB 12 SETFONT
47 205 MOVETO (01-8574-5) SH
219 205 MOVETO (01-8574-5) SH
/NHE 16 14 SETFONT
345.6 813 MOVETO (Facture de consommation) SH
}FSHOW

```



Note: The **FSHOW** command caches the form file automatically. When the form file contains a variable reference, delete the **FSHOW** command. When **FSHOW** is not deleted, only the first instance of the variable object is cached and used for subsequent calls.

## JOB DESCRIPTOR TICKET FILE EXAMPLE

```

%!PS-Adobe-2.0
%%Title: example4.jdt
%%Creator: JYB/RXCH
%%CreationDate: MAR 95
%%For: RPE definition for example4
%%Copyright: (C) 1995 by Rank Xerox AG (RXCH). All right reserved.
%%EndComments

%-----
% set orientation and margins
%-----
PORT                % PORT - Portrait    IPORT - Inverse Portrait
                   % LAND - Landscape    ILAND - Inverse Landscape

0 0 0 0 SETMARGIN    % set all margins to zero

%-----
% set form
%-----
(BVR) SETMEDIA
(example4.frm) SETFORM

%-----
% set RPE fonts
%-----
/F1    /NHE    11 9   INDEXFONT % lignes
/F1a   /NHE    9 9   INDEXFONT % BVR G
/F2    /NHEB   12 12  INDEXFONT % reference
/F3    /NHE    12 9   INDEXFONT % montant
/F4    /NHEB   13 18  INDEXFONT % total
/F5    /NOCRB  10     INDEXFONT % ligne OCRB

%-----
% RPE definition
%-----
4 SETRPEPREFIX
12 BEGINRPE

% Almt rot. Xinit Xdisp Yinit Ydisp Rec.pos. Length Font Color

/PERO RPEKEY
[ 0 0 1603 0 212 0 5 30 /F1 BLACK ]

/REFO RPEKEY
[ 2 0 390 0 410 0 5 20 /F2 BLACK ]

/ADRO RPEKEY
[ 0 0 1440 0 410 50 5 99 /F1 BLACK ]
[ 0 0 1410 0 2890 50 5 99 /F1 BLACK ]
[ 0 0 30 0 3030 48 5 99 /F1a BLACK ]

/DFAO RPEKEY
[ 0 0 465 0 510 0 5 10 /F1 BLACK ]

/DPYO RPEKEY
[ 0 0 465 0 560 0 5 10 /F1 BLACK ]

/DREO RPEKEY
[ 0 0 465 0 640 0 5 10 /F1 BLACK ]

```

```

/CRNO RPEKEY
[ 0 0 465 0 740 50 5 50 /F1 BLACK ]

/LFA0 RPEKEY
[ 0 0 70 0 960 50 5 2 /F5 BLACK ]
[ 0 0 130 0 960 50 7 22 /F1 BLACK ]
[ 1 0 920 0 960 50 29 6 /F1 BLACK ]
[ 1 0 1170 0 960 50 35 9 /F1 BLACK ]
[ 1 0 1420 0 960 50 44 8 /F1 BLACK ]
[ 1 0 1620 0 960 50 52 6 /F1 BLACK ]
[ 1 0 1870 0 960 50 58 9 /F1 BLACK ]
[ 1 0 2100 0 960 50 67 7 /F1 BLACK ]
[ 1 0 2410 0 960 50 74 11 /F3 BLACK ]

/LFA1 RPEKEY
[ 0 0 70 0 960 50 5 24 /F3 BLACK ]
[ 1 0 2410 0 960 50 74 11 /F3 BLACK ]

/MNT0 RPEKEY
[ 0 0 1600 0 2180 0 5 17 /F3 BLACK ]
[ 1 0 2410 0 2190 0 22 11 /F4 BLACK ]
[ 1 0 340 0 2790 0 22 7 /F5 BLACK ]
[ 1 0 460 0 2790 0 30 2 /F5 BLACK ]
[ 1 0 1060 0 2790 0 22 7 /F5 BLACK ]
[ 1 0 1180 0 2790 0 30 2 /F5 BLACK ]

/BRE0 RPEKEY
[ 0 0 1300 0 2800 0 5 32 /F5 BLACK ]
[ 0 0 30 0 2910 0 5 15 /F1 BLACK ]
[ 0 0 30 0 2958 0 20 17 /F1 BLACK ]

/OLIO RPEKEY
[ 0 0 810 0 3290 0 5 42 /F5 BLACK ]
ENDRPE

```

The following figure illustrates the printed output for Example 4.

[illegible]

## Example 5: Mailing Using Database Mode

This section contains an example of a mailing using database mode. The example consists of a print file, a DBM file, and an example of the generated output.

### PRINT FILE EXAMPLE

```
%!
(example5.dbm) STARTDBM
CHER:TITRE:FNAME:NAME:ADRESS1:ADRESS2:ZIP:CITY
Cher:Monsieur:Alain:DUPONT:3, rue de la gare::1200:Genève
Chère:Madame:Martine:BELLEGRAMBE:chez son copain:12, place de la poste:1034:Lausanne
%%EOF
```

### DATA BASE MASTER FILE EXAMPLE

```
%%title: example5.dbm
%!
PORT
/M /NTMR 12 11.3 INDEXFONT % M = medium font
/B /NTMB 12 11.3 INDEXFONT % B = bold font
1200 0 360 0 SETMARGIN
48 SETLSP

1240 3300 MOVETO
(dcxlogo.seg).6 SCALL

1450 2800 MOVETO
M ($$TITRE. $$FNAME. //B$$NAME.) VSUB 0 SHMF M
ADRESS1 SHL
ADRESS2 SHL
ZIP SH ( - ) SH CITY SH

1780 SETCOLWIDTH
360 2300 MOVETO
($$CHER. $$TITRE. $$NAME..) VSUB SHL
NL
(Die welt, in der Sie leben, //B$$CHER. $$TITRE. $$NAME.//M, ist - in Übereinstimmung mit
Ihrer Persönlichkeit - aussergewöhnlich.) VSUB 3 SHP
NL
(So aussergewöhnlich wie der Trembley Brooklyn, ein Automobil der exklusivsten Klasse.) SHJ
(Wobei sich der Begriff Exklusivität sowohl auf das aussergewöhnlich hohe Niveau der) SHJ
(Verarbeitung, des Luxus, als auch auf die Sicherheit und die Fahrleistungen erstreckt.) SHJ
NL
(Und, auf die Dauer gesehen, kommt ein Brooklyn nicht teurer zu stehen als ein) SHJ
(Spitzenprodukt irgend eines Grossherstellers: In 30 Jahren wird Ihr Brooklyn immer) SHJ
(noch modern und fahrtüchtig sein, wie es zwei Drittel der seit 73 Jahren produzierten) SHJ
(Trembleys heute noch sind.) SHL NL
(Wir möchten es Ihnen ermöglichen, während einigen Stunden die welt des Trembley) SHJ
(Brooklyn zu erleben.) SHL
NL
(Im wahrsten Sinne des Wortes. Wo und wann dies Ihnen genehm ist und zwar auf die) SHJ
(angenehmste, einfachste Weise.) SHL
NL
(Wir fahren bei Ihnen vor, Ihre welt und unsere welt begegnen sich. Sie dürfen sich dann) SHJ
(davon überzeugen, dass der Brooklyn im Einklang mit Ihren persönlichen Wertmassstäben) SHJ
(steht.) SHL
```

```
NL
(Es genügt, wenn Sie uns beiliegende Einladungskarte zurücksenden: wir werden uns) SHJ
(nachher um alles kümmern.) SHL
NL
(Lassen Sie sich überraschen. Wir werden Ihnen die Aufmerksamkeit widmen, die Sie) SHJ
(verdienen.) SHL
NL
(Gerne erwarten wir Ihre Antwort und verbleiben inzwischen mit freundlichen Grüßen.) SHJ
NL
NL
(signa.tif) .75 0 ICALL
NL NL NL NL
(Your W. Signature) SHL
(MANAGING DIRECTOR) SHL
PAGEBRK
```

The following figure illustrates the printed output for Example 5.

# xerox

Monsieur Alain **DUPONT**  
3, rue de la gare  
1200 - Genève

Cher Monsieur DUPONT,

Die Welt, in der Sie leben, Cher Monsieur DUPONT, ist - in Übereinstimmung mit Ihrer Persönlichkeit - aussergewöhnlich.

So aussergewöhnlich wie der Trembley Brooklyn, ein Automobil der exklusivsten Klasse. Wobei sich der Begriff Exklusivität sowohl auf das aussergewöhnlich hohe Niveau der Verarbeitung, des Luxus, als auch auf die Sicherheit und die Fahrleistungen erstreckt.

Und, auf die Dauer gesehen, kommt ein Brooklyn nicht teurer zustehen als ein Spitzenprodukt irgend eines Grossherstellers: In 30 Jahren wird Ihr Brooklyn immer noch modern und fahrtüchtig sein, wie es zwei Drittel der seit 73 Jahren produzierten Trembleys heute noch sind.

Wir möchten es Ihnen ermöglichen, während einigen Stunden die Welt des Trembley Brooklyn zu erleben.

Im wahrsten Sinne des Wortes. Wo und wann dies Ihnen genehm ist und zwar auf die angenehmste, einfachste Weise.

Wir fahren bei Ihnen vor, Ihre Welt und unsere Welt begegnen sich. Sie dürfen sich dann davon überzeugen, dass der Brooklyn im Einklang mit Ihren persönlichen Wertmassstäben steht.

Es genügt, wenn Sie uns beiliegende Einladungskarte zurücksenden. Wir werden uns nachher um alles kümmern.

Lassen Sie sich überraschen. Wir werden Ihnen die Aufmerksamkeit widmen, die Sie verdienen.

Gerne erwarten wir Ihre Antwort und verbleiben inzwischen mit freundlichen Grüßen.

*Your Signature*

Your W. Signature  
MANAGING DIRECTOR

## Example 6: Bank Statement Using Dynamic Boxes and Data-Driven Graphics

This section contains an example of a bank statement using RPE line mode with dynamic boxes and DDG. The example consists of a print file, a JDT, and the printed output.

### PRINT FILE EXAMPLE

```
%!
(example6.jdt) STARTLM
IACS0:PORTFOLIO OVERVIEW
  ACS1:Asset Allocation Summary
  ACS2:::2 JUL 1997:Asset Class:::SGD
  ACS3:BOND, NOTE:::925,647.30:41.85
  ACS3:VARIOUS:::25,802.48:11.17
  ACS3:CASH:::37,586.28:32.70
  ACS3:SHARE:::1,222,538.43:45.28
  ACS4:TOTAL:2,211,574.49:100.00
  ACS1:Currency Allocation Summary
  ACS2:Market Value:FX Rate:::Currency:local currency:(LCU/SGD):SGD
  ACS3:AUSTRALIAN DOLLAR:33,730.00:1.0760:36,293.48:3.64
  ACS3:SWISS FRANCS:118,704.50:0.9753:115,772.50:5.23
  ACS3:DEUTSCHE MARK:1,297,995.69:0.8189:1,062,928.67:23.06
  ACS3:DANISH KRONER:1,032,533.33:0.2151:222,097.92:10.04
  ACS3:FRENCH FRANCS:513,365.00:0.2429:124,696.36:5.64
  ACS3:HONG KONG DOLLARS:186,400.00:0.1845:34,390.80:2.56
  ACS3:YEN (JAPAN):15,343,050.00:0.0125:191,481.26:8.66
  ACS3:MALAYSIAN RINGGIT:37,200.00:0.5662:21,062.64:1.95
  ACS3:NEW ZEALAND DOLLARS:72,061.98:0.9695:69,864.09:3.16
  ACS3:PHILIPPINE PESOS:363,000.00:0.0543:19,707.27:1.89
  ACS3:SINGAPORE DOLLAR:62,358.40:::62,358.40:3.82
  ACS3:US DOLLAR:175,555.24:1.4293:250,921.10:11.35
  ACS4:TOTAL:2,211,574.49:100.00
ITIT0:TIME WEIGHTED SEGMENT PERFORMANCE
  SEM0:Period from 1 January 1997 to 2 July 1997
  SEM1:Asset Class:Capital Flows SGD:Total Gains/Losses SGD:Total Earnings
SGD:Performance
  SEM3:CASH:1,220.18:1,400.26:0.00:3.80
  SEM3:BOND, NOTE:0.00:3,737.23:197.24:5.38
  SEM3:SHARE:-2150.26:47,098.98:215.26:-3.90
  SEM3:FIXED INCOME:16,313.61:19,423.00:3,188.89:8.06
  SEM3:EQUITIES:-40,355.07:17,958.50:1,828.57:9.83
  SEM3:EQUITY FUNDS:-16,409.67:4,865.38:583.29:2.71
  SEM3:VARIOUS:0.00:708.84:0.00:2.82
ITIT0:TIME WEIGHTED GROSS PERFORMANCE
  PER0:Period from 1 January 1997 to 31 July 1997
  PPA0:TOTAL ASSET VALUE END OF PERIOD:2,211,574.49:SGD
  PPA0:TOTAL ASSET VALUE START OF PERIOD:2,208,526.77:SGD
  PPA0:DEPOSITS:0.00:SGD
  PPA0:WITHDRAWALS:0.00:SGD
  PPA0:CUSTODY/MANAGEMENT FEES:::SGD
  PPA0:WITHHOLDING TAX:0.00:SGD
  PPA1:VALUATION CHANGE:3,047.72:SGD:PERFORMANCE:0.09 %
  PPA2:PERFORMANCE COMPONENTS:
```

```

PPA0:REALIZED GAINS/LOSSES SECURITIES:0.00:SGD
PPA0:Unreal. GAINS/LOSSES SECURITIES:6,544.05:SGD
PPA0:REALIZED GAINS/LOSSES FX:0.00:SGD
PPA0:UNREAL. GAINS/LOSSES FX:-4,913.75:SGD
PPA0:DIVIDENDS AND INTEREST:215.26:SGD
PPA0:CHANGE IN ACCR. INTEREST:197.25:SGD
PPA0:CUSTODY/MANAGEMENT FEES::SGD
PPA0:WITHHOLDING TAX:0.00:SGD
PPA0:EXTRAORDINARY EXPENSES:0.00:SGD
PPA0:EXTRAORDINARY INCOME:0.00:SGD
PPA1:VALUATION CHANGE:3,047.72:SGD
PPA4:0.15:January
PPA4:0.10:February
PPA4:0.16:MarchPPA4:0.18:April
PPA4:0.06:May
PPA4:0.14:June
PPA4:0.09:July
PPA5
%%EOF

```

## JOB DESCRIPTOR TICKET FILE EXAMPLE

```

%!PS-Adobe-2.0
%%Title: example6.jdt
%%Creator: JYB/XCH
%%CreationDate: Jan. 99
%%Copyright: (C) 1998 Xerox Corporation. All Rights Reserved.
%%EndComments

/ANSI SETPCC
%-----
% set orientation
%-----
%2480 3500 SETPAGESIZE
ILAND

%-----
% top bottom left right margins
%-----
0 0 0 0 SETMARGIN
130 65 SETGRID

%-----
% form and page numbering
%-----
{ ILAND 3110 2273 MOVETO RED SETTXC (xdlogo.tif) .66 0 ICALL } SETFORM
(Page #) 1 1750 100 7 SETPAGENUMBER

%-----
% tests
%-----
/SEGPREF 1 5 1 4 /eq (SEM3) SETPCD
/IF_PERF 5 /FN /ne () SETRCD

%-----
% set RPE fonts
%-----
/F1 /NHEB 16 INDEXFONT
/F3 /NHEB 13 INDEXFONT
/F3B /NHEB 12 9 INDEXFONT
/F4 /NHE 10 INDEXFONT
/F4B /NHEB 10 INDEXFONT
/F7 /NHE 7 9 INDEXFONT
/F8 /NHE 9 INDEXFONT

%-----
% RPE definition
%-----

4 SETRPEPREFIX
20 BEGINRPE

% Almt      rot. Xinit Xdpl Yinit Ydpl Pos. Length Font Color
/TIT0 RPEKEY
[ 2          0 1750 null 450 0 1 /FN /F1 BLACK]

/ACS0 RPEKEY
[ 2          0 1750 null 450 0 1 /FN /F1 BLACK]
/ACS1 RPEKEY
[ {SCALL}    0 250 null 0 90 0 (BLUEBOX) /F1 BLACK]
[ {SCALL}    0 250 null 0 0 0 (LINEHOR) /F1 BLACK]
[ 0          0 270 null 0 60 1 /FN /F3 WHITE]
[ {SCALL}    0 250 null 0 20 0 (LINEVERT) /F1 BLACK]

```

```

/ACS2 RPEKEY
[ 1 0 1330 null 0 60 1 /FN /F4B BLACK]
[ 1 0 1600 null 0 0 2 /FN /F4B BLACK]
[ 1 0 2030 null 0 0 1 /FN /F4B BLACK]
[ 0 0 1970 null 0 0 3 /FN /F4B BLACK]
[{SCALL} 0 250 null 0 0 0 (LINEVERT) /F1 BLACK]
[ 0 0 270 null 0 50 4 /FN /F4B BLACK]
[ 1 0 1330 null 0 0 5 /FN /F4B BLACK]
[ 1 0 1600 null 0 0 6 /FN /F4B BLACK]
[ 1 0 2030 null 0 0 7 /FN /F4B BLACK]
[ 1 0 2250 null 0 0 0 (%) /F4B BLACK]
[{SCALL} 0 250 null 0 12 0 (LINEHOS) /F1 BLACK]
[{SCALL} 0 250 null 0 0 0 (LINEVERT) /F1 BLACK]

/ACS3 RPEKEY
[ 0 0 270 null 0 50 1 /FN /F4 BLACK]
[[(.)6] 0 1270 null 0 0 2 /FN /F4 BLACK]
[[(.)6] 0 1500 null 0 0 3 /FN /F4 BLACK]
[[(.)6] 0 1970 null 0 0 4 /FN /F4 BLACK]
[[(.)6.1] 0 2190 null 0 0 5 /FN /F4 BLACK]
[{SCALL} 0 250 null 0 0 0 (LINEVERT) /F1 BLACK]

/ACS4 RPEKEY
[{SCALL} 0 250 null 0 40 0 (GRAYBOX) /F3B BLACK]
[{SCALL} 0 250 null 0 0 0 (LINEVERT) /F1 BLACK]
[ 0 0 270 null 0 46 1 /FN /F4 BLACK]
[[(.)6] 0 1970 null 0 0 2 /FN /F4B BLACK]
[[(.)6] 0 2190 null 0 0 3 /FN /F4B BLACK]
[{SCALL} 0 250 null 0 15 0 (LINEVERT) /F1 BLACK]
[{SCALL} 0 250 null 0 0 0 (LINEHOR) /F1 BLACK]
[{SCALL} 0 2790 null 0 0 0 (PIE1) /F8 BLACK]

%-----
[ 1 0 2190 null 0 0 3 /FN /F4 BLACK]
[ 1 0 2790 null 0 0 4 /FN /F4 BLACK]
[ 1.1 0 3230 null 0 0 5 /FN /F4B BLACK]
[{SCALL} 0 0 null 0 10 0 (LIVERGR) /F1 BLACK]
[{SCALL} 0 250 null 0 0 0 (LINEVERT) /F1 BLACK]

%-----

/SEM0 RPEKEY % Segment performance blue title line
[{SCALL} 0 250 null 470 75 0 (PERFRAME) /F1 BLACK]
[{SCALL} 0 250 null 0 0 0 (BLUEBOX) /F1 BLACK]
[ 0 0 270 null 0 55 1 /FN /F3 WHITE]

/SEM1 RPEKEY % Header of the segment performance report
[ 0 0 260 null 470 120 1 /FN /F4B BLACK]
[ 0 0 1010 null 0 0 2 /FN /F4B BLACK]
[ 0 0 1610 null 0 0 3 /FN /F4B BLACK]
[ 0 0 2210 null 0 0 4 /FN /F4B BLACK]
[ 0 0 2810 null 0 0 5 /FN /F4B BLACK]
[{SCALL} 0 0 null 0 10 0 (LIVERGR) /F1 BLACK]
[{SCALL} 0 250 null 0 0 0 (LINEHOR) /F1 BLACK]
[{SCALL} 0 250 null 0 0 0 (LINEVERT) /F1 BLACK]

/SEM3 RPEKEY % Simple line of the segment performance report
[ 0.2 0 260 null 470 65 1 /FN /F4 BLACK]
[ 1 0 1590 null 0 0 2 /FN /F4 BLACK]

/PER0 RPEKEY % Performance blue title line
[{SCALL} 0 250 null 470 75 0 (PERFRAME) /F1 BLACK]
[{SCALL} 0 250 null 0 0 0 (BLUEBOX) /F1 BLACK]
[ 0 0 270 null 0 55 1 /FN /F3 WHITE]

```

```

/PPA0 RPEKEY % left side of the page of the performance report
[ 0 0 270 null 620 75 1 /FN /F4 BLACK]
[ 1 0 1550 null 0 0 2 /FN /F4 BLACK]
[ 1 0 1700 null 0 0 3 /FN /F4 BLACK]
/PPA1 RPEKEY % left side of the page of the performance report (subtotal)
[ 0 0 270 null 0 75 1 /FN /F4B BLACK]
[ 1 0 1550 null 0 00 2 /FN /F4B BLACK]
[ 1 0 1700 null 0 00 3 /FN /F4B BLACK]
/IF PERF
[ {SCALL} 0 1830 null 0 0 0 (GRAYBOX8) /F3B BLACK]
[ 0 0 1850 null 0 0 4 /FN /F4B BLACK]
[ 1 0 3150 null 0 0 5 /FN /F4B BLACK]
/ENDIF

/PPA2 RPEKEY % Subtitle of the left side of the performance report.
[ 0 0 270 null 0 200 1 /FN /F4B BLACK]
[ 0 0 1650 null 0 0 2 /FN /F4B BLACK]
/PPA4 RPEKEY % performance data
[ 0.1 0 -500 null 0 0 1 /FN /F4B BLACK]
[ 0.2 0 -500 null 0 0 2 /FN /F4B BLACK]

/PPA5 RPEKEY % performance curve
[ {SCALL} 0 1940 null 2150 null 0 (CRV1) /F8 BLACK]

ENDRPE

/BLUEBOX
{0 0 3000 80 FBLUE DRAWB} XGFRESDEF

/GRAYBOX
{0 0 2010 60 LT DRAWB} XGFRESDEF

/GRAYBOX8
{0 -10 1418 -60 LT DRAWB} XGFRESDEF

/LINEHOS
{0 0 2010 4 FBLACK DRAWB} XGFRESDEF
/LINEHOR
{-4 0 3008 4 FBLACK DRAWB} XGFRESDEF

/LINEVERT
{-4 0 4 -80 FBLACK DRAWB
3000 0 4 -80 FBLACK DRAWB
} XGFRESDEF
/PERFRAME
{-4 0 3008 -4 FBLACK DRAWB
-4 -1758 3008 -4 FBLACK DRAWB
-4 0 4 1758 FBLACK DRAWB
3004 0 -4 1758 FBLACK DRAWB
} XGFRESDEF

/LIVERGR
{1000 -3 3 -100 LT DRAWB
1600 -3 3 -100 LT DRAWB
2200 -3 3 -100 LT DRAWB
2800 -3 3 -100 LT DRAWB
} XGFRESDEF

/PIE1
{ 0 380 [/15
/SpotOffset -1.35
/ExtraSpace 2
/3DAngle .25
/FitSpace 8
] DRAWPIE
} XGFRESDEF

```

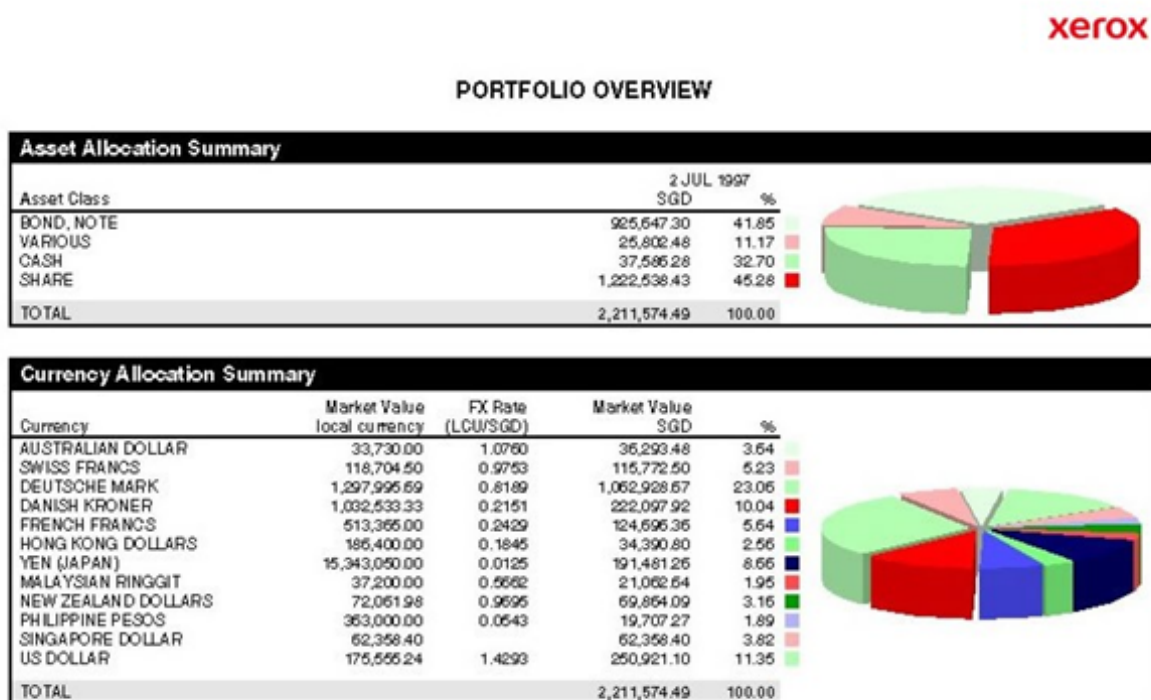
```

{ IF SEGPREF
{ ORITL
253 /SEM3 8 /V RPEPOS 2994 3 LT DRAWB
F7 550 2200 MOVETO
/VAR.BARH 1620 /SEM3 0 /VD RPEPOS sub SETVAR% compute available height
0 2400 VAR.BARH [/67
/BGColor XLBLUE
/ColorTable [LGREEN]
/BarSpace .25
/Fitspace 0
] DRAWBAR
} ENDIF
} ENDPAGE

/CRV1
{ 0 1250 800 [/67
/BGColor XLBLUE
/ColorTable [LGREEN]
/3DThickness .5
/Fitspace 0
] DRAWCRV
} XGFRESDEF

```

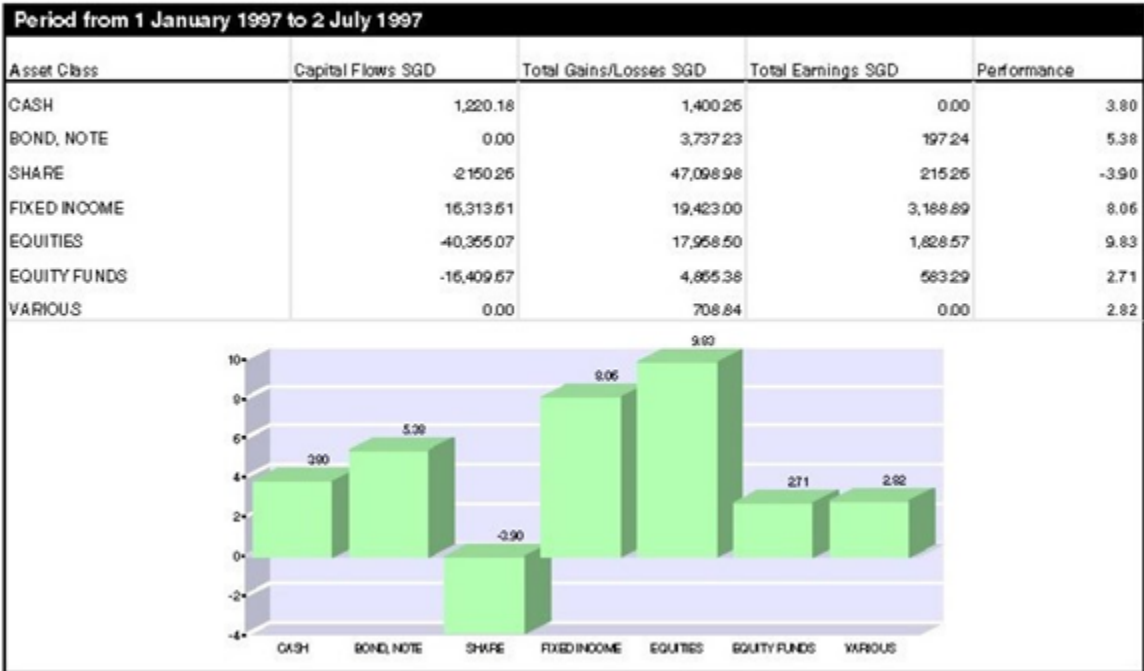
The following example illustrates printed output for a bank statement that contains dynamic boxes and pie chart DDGs.



The following example illustrates printed output for a bank statement that contains dynamic boxes and bar chart DDGs.

xerox

## TIME WEIGHTED SEGMENT PERFORMANCE

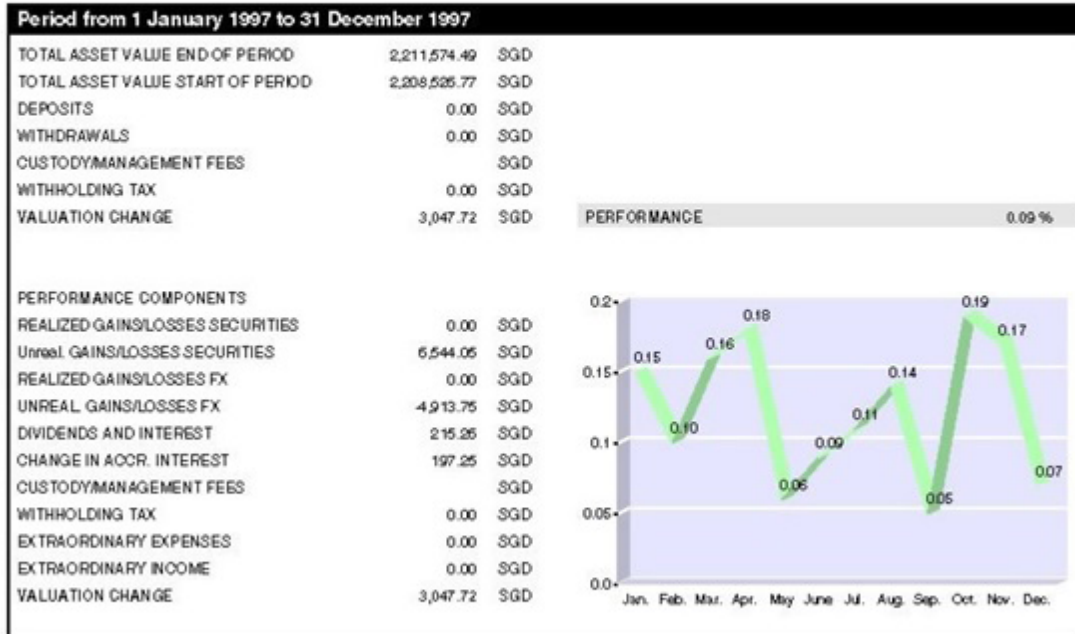


Page 3

The following example illustrates printed output for a bank statement that contains dynamic boxes and curve chart DDGs.

xerox

## TIME WEIGHTED GROSS PERFORMANCE





# Standard Lists, Tables, Keys, and Attributes

This chapter contains:

Standard Font Lists and Encoding Tables .....	164
Standard GEPkeys, PATkeys, BATkeys, and Colorkeys .....	167
Predefined PCC Tables .....	168
Predefined Multi-Up Definitions.....	169
Predefined Multiple-Byte Encoding Table .....	170

Information about locating the lists, tables, keys, attributes, and other predefined information provided with VI Compose can be found in these sections:

- Standard font lists and encoding tables
- Standard GEPkeys, PATkeys, BATkeys and Colorkeys
- Predefined PCC tables
- Predefined Multi-Up definitions
- Predefined multi-byte encoding table

## Standard Font Lists and Encoding Tables

Font lists and encoding tables can be found in the following subdirectories of `xgf/encoding`:

<code>/fontlist</code>	Re-encoded font list file
<code>/mac</code>	Macintosh character set
<code>/nullfl</code>	PostScript encoded font list file
<code>/pc8</code>	pc-8 encoding table file
<code>/pcsun</code>	pcsun encoding table file
<code>/sun8</code>	ISO Latin 1 encoding table
<code>/utf8</code>	UTF-8 encoding table
<code>/win1252</code>	win1252 encoding table



Note: When coded in a job, the **SETENCODING** statement is in effect for that job only. After the job ends, the **SETENCODING** statement returns to the default setting, as specified in the `xgf.def` file.

When most of the jobs use the **SETENCODING** statement, replace the default encoding table `pcsun` as the default by invoking another table permanently. To invoke another table permanently, edit the **SETENCODING** statement in `xgf.def`, which is invoked by `/usr/xgf/src/xgfunix.run` or `\xgf\src\xgfdos.run`.

### RE-ENCODED FONT LIST FILE

The VIPP® re-encoded font list in `xgf/encoding/fontlist` contains the following:

- VIPP® font key names for standard PostScript font names
- VIPP® font key names for additional DocuPrint font names and their substitute font names
- VIPP® font key names for purchased fonts and their substitute font names

The information in the tables is found in `xgf/encoding/fontlist`. The default re-encoding table is specified using the **SETENCODING** command in `xgf/src/xgf.def`. The file is invoked in `/usr/xgf/src/xgfunix.run` and `\xgf\src\xgfdos.run`.

### MACINTOSH CHARACTER SET

The encoding table `xgf/encoding/mac` provides information about the encoded character set for key positions 128–255 for Macintosh operating systems. To invoke the encoding table permanently, edit the fontlist **SETENCODING** statement in `xgf/src/xgf.def`, or, on a case-by-case basis in custom **SETENCODING**s.

Only those characters in the PostScript Standard Latin Character Set are encoded.

### POSTSCRIPT-ENCODED FONT LIST FILE

A font list is provided in `xgf/encoding/nullfl` that is associated with the null encoding table that uses **SETENCODING** in `xgf/src/xgf.def`. Null encoding indicates that the original encoding of the PostScript font

is retained in the new font, sometimes referred to as a non re-encoded font. The non re-encoded font is used to assign a shorter name to fonts that have a non-Latin character set.

VIPP® FONT KEY	POSTSCRIPT FONT NAME
NSY	Symbol
NZDB	Zapf Dingbats

### PC-8 ENCODING TABLE FILE

The PC-8 table provided in `xf/encoding/pc8` provides information about the encoded character set for key positions 128–255 for DOS operating systems. To invoke the PC-8 encoding, use a **SETENCODING** command. Only those characters that are present in the PostScript Standard Roman Character Set are encoded.

### PCSUN ENCODING TABLE FILE

The pcsun encoding table in `xf/encoding/pcsun` provides information about the encoded character set for key positions 128–255. Key positions 128–175 replicate the corresponding key positions in the pc8 font-encoding table. Key positions 176–255 replicate the corresponding key positions in the sun8 font-encoding table.

### ISO LATIN 1 ENCODING TABLE

The ISO Latin 1 encoding table that is provided in `xf/encoding/sun8` contains the following line of code:

```
ISOLatin1Encoding aload pop
```

The code calls the PostScript ISO Latin1 Encoding array and is invoked using a **SETENCODING** command. The array provides information about the encoded character set for key positions 128–255 for Latin text fonts that are produced by Adobe. The PostScript ISO Latin1 Encoding array is the character set used by most Windows and UNIX systems.

### UTF-8 ENCODING TABLE

The purpose of UTF8 encoding is to allow the use of base PostScript fonts such as Courier, Helvetica, Times-Roman, and so on, to print UTF-8 data streams.

UTF-8 encoding allows the following character code ranges:

<b>1 byte</b>	00-7F
<b>2 bytes</b>	C080-DFBF
<b>3 bytes</b>	E08080-EFBFBF
<b>4 bytes</b>	F0808080-F7BFBFBF

The ranges contain thousands of characters. Only those character codes that can be mapped with a glyph in the base font, generally using the Standard Roman Character Set, are imaged. Other character codes default to the question mark (?). The following are examples of UTF-8 encoding:

```
16#C3A8 /grave /acute /circumflex /edieresis
16#C2A3 /sterling
16#C2A5 /yen
16#E282AC /Euro
```

Character codes in the 4-byte range of the UTF-8 specifications require hexadecimal strings. For example:

```
<F08092A0> /character_name.
```

To re-encode a base font for UTF-8 data streams, use a UTF-8 encoding table with the **SETENCODING** command, as in the following example:

```
[ /CR-UTF8 /Courier
/HE-UTF8 /Helvetica
/TM-UTF8 /Times-Roman
] (utf8) SETENCODING
```

Using -UTF8 in the VIPP® font name is not mandatory but is recommended to benefit from the character boundary recognition in VIPP® commands that are related to the strings RPE/GETFIELD/SETRCD/ SETPCD/GETINTV.

-UTF8 is registered as a guess string in cjk.def.

## WIN1252 ENCODING TABLE

The VIPP® default font-encoding table, `xf/encoding/win1252`, provides information about the encoded character set for key positions 128–255.

## Standard GEPkeys, PATkeys, BATkeys, and Colorkeys

The Standard VIPP® Colorkeys, Graphic Element Property keys (GEPkeys), and Pattern keys (PATkeys) are provided in the `xgf/src/xgf.gep` file. The file is invoked in `/usr/xgf/src/xgfunix.run` and `\xgf\src\xgfdos.run`.

### GRAPHIC ELEMENT PROPERTY KEYS AND COLORKEYS

To generate **GEPkey** samples, print the `samgep.ps`, `samgepb.ps`, `samgepg.ps`, and `samgepr.ps` files located in `xgf/demo`.

### PATTERN KEYS

To generate **PATkey** samples, print the `sampat.nm` file in `/xgf/demo`.

### STANDARD BACKGROUND ATTRIBUTES

The Standard VIPP® Background Attributes keys (BATkeys) are used to print a specific background for each character printed when the BATkey is in effect. BATkeys are provided in the `xgf/src/xgf.bat` file. To see both black and white and color BATkeys, consult the `xgf.bat` file. The file is invoked in `/usr/xgf/src/xgfunix.run` and `\xgf\src\xgfdos.run`.

To generate BATkey samples, print the `sambat.ps`, `sambatb.ps`, `sambatg.ps`, and `sambatr.ps` files in `xgf/demo`.

### SOLID COATED AND UNCOATED CUSTOM COLORS

VI Compose supports solid coated and uncoated colors for color-enabled printing systems. The custom Colorkeys are defined in the files `xgf/src/scoat.cck` for jobs on coated paper, and `sucoat.cck` for jobs on uncoated paper. Because of chemical differences between coated and uncoated papers, printed colors can vary from one paper to the other, even when using the same color setting.

To use VIPP® custom colors, include the custom Colorkeys in VIPP® jobs as with any Colorkey defined in `xgf.gep`.

## Predefined PCC Tables

Predefined PCC tables are provided in the `xgf/src/xgf.pcc` file. The file is invoked in `/usr/xgf/src/xgfunix.run` and `\xgf\src\xgfdos.run`. A PCC table is invoked using a **SETPCC** command in a JDT.

In addition, the IBM STD2 FCB is provided in a **SETVFU** definition, and a default **SETVFU** definition is invoked. Job-specific **SETVFU** commands are required in the corresponding JDTs.

## Predefined Multi-Up Definitions

Predefined Multi-Up definitions are provided in the `xgf/src/xgf.mup` file. This file is invoked in `/usr/xgf/src/xgfunix.run` and `\xgf\src\xgfdos.run`.

A pre-defined Multi-Up definition is used to replicate the original page image many times on a physical page. For example, the pre-defined **TWOUP** Multi-Up definition prints two logical pages on one physical page. This feature is used often in sysout reports, to print postcards, or for label applications.

To invoke a Multi-Up definition, use the **SETMULTIUP** command in a JDT.

## Predefined Multiple-Byte Encoding Table

The VI Compose configuration file, `xf/src/cjk.def` contains information that is necessary to produce VIPP® jobs, using multiple-byte fonts. The configuration file includes the following tables:

- FontName to Encoding mapping table
- Wrapping rules character table. Kinsoku rules are used for Japanese fonts.
- Vertical-to-horizontal character tables

# Printing with VI Compose

This chapter contains:

Decomposition Services .....	172
Using VI Compose on Specific Printers .....	176
VI Compose and FreeFlow Makeready .....	187
LCDS Migration .....	194

*Printing with VI Compose* contains information about these print-related functions:

Decomposition services	Describes how to use the preprocessing capabilities on the DocuPrint NPS printers and FreeFlow Print Server controllers
Using VI Compose on specific printers	Provides printer-specific hints and tips
VI Compose and FreeFlow Makeready	Describes the interaction of FreeFlow Makeready Suite and VI Compose functions
LCDS migration	Provides information helpful in the migration of LCDS data files. LCDS data files can be printed to a VIPP®-enabled Xerox Production Printer, often without changing the original mainframe application.

## Decomposition Services

VI Compose can take advantage of the printer preprocessing capabilities that are added to the DocuPrint NPS and FreeFlow Print Server. VI Compose uses Decomposition Services to print preprocessed documents, using the **RUNDD** command. To print documents that are referenced as forms or images, use the **SETFORM**, **SETBFORM**, and **ICALL** commands.

The default **SETFPATH** and **SETIPATH** commands, which are in the `/usr/xgf/src/xgfunix.run` file, provide the path to the directory that contains the forms or TIFF images that the Decomposition Services creates.

- For the DocuPrint NPS, the path is `/var/db/forms`.
- For the FreeFlow Print Server, the path is provided using the GUI menus.

The implementation of Decomposition Services for the FreeFlow Print Server dynamically creates a new directory for each document that is processed. Only the path to the top forms directory can be provided.

### REFERENCING DECOMPOSED FORMS USING A GENERIC METHOD

This section contains information on how to reference forms that are decomposed on any system.

#### Syntax

```
{ ( document_name) RUNDD } SETFORM
{ ( document_name) RUNDD } SETBFORM
```

#### Where:

<b>document_name</b>	Name of the file submitted to Decomposition Services. Requires a single-page document.
----------------------	---

#### Example

The following example selects a previously single-page decomposed file, `letter.prn`, as a form:

```
{ (letter.prn) RUNDD } SETFORM
```

### REFERENCING FORMS DECOMPOSED IN DOCUPRINT NPS

This section contains information on how to reference forms decomposed in DocuPrint NPS. Both TIFF and Byte Compress formats are supported.

#### Syntax

```
(document_name.pnnnn.ps) SETFORM
(document_name.pnnnn.ps) SETBFORM
```

#### Where:

<b>document_name</b>	is the name of the file submitted to Decomposition Services
<b>nnnn</b>	is the four-digit page number.

## Example

This example selects the first page of a previously decomposed file called `letter` as a form.

```
(letter.p0001.ps) SETFORM
```

## REFERENCING FORMS DECOMPOSED IN THE FREEFLOW PRINT SERVER

This section contains information on how to reference forms that are decomposed in the FreeFlow Print Server.

### Syntax

```
{0 0 MOVETO  
(document_name_dir/document_name.pnnnnnnn.tif) 1 90 ICALL} SETFORM or SETBFORM
```

### Where:

document_name	The name of the file submitted to Decomposition Services.
nnnnnnnn	The eight-digit page number.

## Example

The following example selects the first page of a previously decomposed file called `letter` as a form.

```
{ 0 0 MOVETO (letter_dir/letter.p00000001.tif) 1 90 ICALL} SETFORM
```

## REFERENCING DECOMPOSED IMAGES IN DOCUPRINT NPS DEVICES

This section contains information on how to reference images that are decomposed in DocuPrint NPS devices.

### Syntax

```
document_name.pnnnn.c.tif, scaling rotation ICALL
```

### Where:

document_name	The name of the file submitted to Decomposition Services.
c	is either b for black images or h for highlight images.
nnnn	The four-digit page number.

## Example

The following example prints the TIFF image that represents the first page of a previously decomposed file called `letter`. The TIFF image is printed at the current position, in the original size, with no rotation.

```
0 0 MOVETO  
(letter.p0001.b.tif) 1 90 ICALL
```



Note: Decomposition Services rotate images for optimization. To print the image in the original orientation and for the best performance, in the **ICALL** command, use a value of 90.

The **ICALL** command can reference only TIFF images. Byte Compress format is not supported.

## REFERENCING DECOMPOSED IMAGES IN THE FREEFLOW PRINT SERVER

This section contains information on how you reference images that were decomposed in the FreeFlow Print Server.

### Syntax

`document_name_dir/document_name.pnnnnnnnn.tif, scaling rotation ICALL`

### Where:

<code>document_name</code>	The name of the file submitted to Decomposition Services.
<code>nnnnnnnn</code>	The eight-digit page number.

### Example

The following example prints the TIFF image that represents the first page of a previously decomposed file called `letter`. The TIFF image is printed at the current position, in the original size, with no rotation.

```
0 0 MOVETO
(letter_dir/letter.p00000001.tif) 1 90 ICALL
```



Note: Decomposition Services rotate images for optimization. To print images in the original orientation and for the best performance, in the **ICALL** command, use a value of 90.

## DECOMPOSITION SERVICES HINTS AND TIPS

This section contains Decomposition hint and tips.

FreeFlow Print Server and DocuColor systems support **CACHE**. Consider using the **CACHE** command for its efficiency.

### Decomposition document in the Data Base Master

Use one of these commands to call the decomposition document into the Data Base master:

- **RUNDD**: When you call the Decomposition document using the **RUNDD** command, variable data is merged

```
%!
PORT
X1 Y1 MOVETO VAR1 SHL
X1 Y1 MOVETO VAR1 SHL
(decomp_doc.ps) RUNDD
```

on the first page only, as in the following example:

- **ICALL**: You can call all individual pages using the **ICALL** command. The variable data can be merged on any

```
%!
PORT (decomp_doc.ps.p0001.b.tif) 1 90 ICALL
X1 Y1 MOVETO VAR1 SHL
PAGEBRK
(decomp_doc.ps.p0002.b.tif) 1 90 ICALL
X2 Y2 MOVETO VAR2 SHL
PAGEBRK
(decomp_doc.ps.p0003.b.tif) 1 90 ICALL
PAGEBRK
X3 Y3 MOVETO VAR3 SHL
X4 Y4 MOVETO VAR4 SHL
(decomp_doc.ps.p0004.b.tif) 1 90 ICALL
PAGEBRK
```

page, as in the following example:

- **RUNDD** and cycle forms: To define the form for each page that is merged with variable data, use the cycle form syntax, as in the following example:

```

%!
PORT
[ {X1 Y1 MOVETO VAR1 SHL}    % variables for page 1
  {X2 Y2 MOVETO VAR1 SHL}    % variables for page 2
  null                       % nothing on page3
  {X3 Y3 MOVETO VAR3 SHL}    % variables for page 4
  {X4 Y4 MOVETO VAR4 SHL}    % variables for page 5
  null                       % nothing on page 6
  null                       % nothing on page 7
  null                       % nothing on page 8
] SETFORM
(decomp_doc.ps) RUNDD

```

When you have a large job, code the cycle **SETFORM** sequence in a separate JDT. To prevent excessive memory consumption, before the **STARTDBM** line, call **SETFORM** using **SETJDT** in the database file.

### Decomposition form performance

All decomposition forms are formatted with the correct scale and rotation. The format provides the best printer performance available, assuming that you are using the forms at their original sizes.

### Decomposition TIFF file placement

The origin of the TIFF files that are obtained using Decomposition Services is at the bottom-left corner, in portrait orientation. A 90degree rotation for the files is required. By default, **ORIBL** is the bottom-left corner, at position 0,0. When you specify **ORITL**, it is at position 0,3300 for US Letter paper, and at position 0,3500 for A4 paper.

### Importing forms from the NPS decomposition directory

Images do not have to be called when using the decomposition forms as VIPP® forms on the DocuPrint NPS device. To perform this task, add `/var/db/forms` in the **SETFPATH** list and use the **SETFORM** command. For example, to submit a single-page file called `form1.ps` to the Decomposition Services, use the following syntax: `(form1.ps.p0001.ps) SETFORM`

The syntax is required when using Byte Compression (BC), because the **ICALL** command cannot process BC files.

## Using VI Compose on Specific Printers

This section provides this information:

- [VIC considerations for iGen](#)
- [VIC considerations for DocuPrint NPS](#)
- [VIC considerations for FreeFlow Print Server](#)
- [VIC considerations for DocuColor](#)
- [VIC considerations for the office environment](#)
- [VIPP command differences](#)



Note: Any limitations that you encounter while using VIPP® are due to the functions and capabilities of the printer and the implementation of the PostScript interpreters.

### VIC CONSIDERATIONS FOR IGEN

When you specify media on an iGen device, using **setpagedevice** and **SETMEDIA** commands, problems arise because the iGen device matches the Stock Name, but does not match PostScript and VI Compose, because they do not specify a stock name.

To avoid this problem, use the name `Unspecifiedxxxx`. The PSIP requires that all stocks have a unique name in the stock database. Create the unique name on the PSIP, where `xxxx` is user-defined. The FreeFlow Print Server matches the stock name on the first 11 characters, so `xxxx` is not recognized. On the FreeFlow Print Server queue and job, leave all the stock attributes at the default settings, especially the stock name, which defaults to `unspecified`.

You can select the stock with the **SETMEDIA** command on an iGen device. Refer to the example shown.

First, define the media in PSIP. The following example shows two media settings:

```
UnspecifiedCMMW CMMW210G Blue 5 210 Coated 2 sides Gloss Long
```

```
Unspecified Plain Blue 5 120 Coated 2 sides Gloss Long
```

Next, set the queue to use unspecified stock as the default media from the stock list.

Finally, set the corresponding information in the VIPP® application:

```
(Plain::120) SETMEDIA  
(CMMW210G::210) SETMEDIA
```

### VIC CONSIDERATIONS FOR DOCUPRINT NPS

This section provides information specific to the DocuPrint NPS printer series, it includes:

- [Production and Demonstration mode](#)
- [Preprocessing capabilities](#)
- [DocuPrint NPS XGFNub](#)

## Production and Demonstration mode

VI Compose delivered on the DocuPrint NPS is enabled for VIPP® Demonstration mode. This mode allows you to use the full set of VIPP® features on the DocuPrint NPS. Documents larger than 200 pages abort at page 200 and the Selected pages: 1 200 message displays. Refer to [FreeFlow VI Compose Licensing and Installation](#) for more information.

To use VIPP® for production printing, you must activate Production mode by purchasing and installing a device-specific VIPP® license string on the DocuPrint NPS controller. For installation instructions refer to [VI Compose Installation on DocuPrint NPS Devices](#).

## Preprocessing capabilities

Printer preprocessing capabilities have been added to the DocuPrint NPS by Decomposition Services. Refer to [Decomposition services](#) for more information.

## DocuPrint NPS XGFNub

XGFNub is a DocuPrint NPS extension that allows you to execute PostScript and VIPP® commands at the start of any VIPP® or PostScript document. Multiple sets of commands can be enabled. The choice of command sets is determined by the DocumentFormat attribute

### Example:

```
(job1.jdt) STARTLM
```

is executed at the start of the main data file. In this example, this set of commands is called job1. The following text is placed in the `/usr/xgfc/mislib/job1.ini` file:

In this example, the VIPP® command

```
%!  
% custom.ini  
(job1.jdt) STARTLM
```

The main data file is sent to DocuPrint NPS, setting the DocumentFormat attribute to `XGF/job1`. The DocumentFormat attribute is set using a virtual printer, the `lpr -C` switch, or any other attribute setting method available.

New sets of commands can be installed at any time by creating new files called `/usr/xgfc/mislib/xxx.ini`, where `xxx` is the name of the command set. The `.ini` extension is mandatory.

In general, all lowercase characters should be used in place of `xxx`. To invoke a command set named `xxx`, set the DocumentFormat attribute to **XGF/xxx**.

Although XGFNub was designed to run with VIPP® as a means to issue initial VIPP® commands, its use is not restricted to VIPP®. It can also be applied to any PostScript program.

## FreeFlow Print Server print queue

FreeFlow Print Server also has a feature to enable a VIPP® start command and associate it with a print queue. Data files sent to this print queue use the VIPP® start commands associated with the printer. Refer to the FreeFlow Print Server user guide for more information on this feature.

## Color rendering on DocuPrint NPS for GEPkeys

To obtain the best color rendering on GEPKeys on a DocuPrint NPS, set the DefaultHighlightColorRendering option on the DocuPrint NPS to **Color Tables**.

## Duplex printing with VIPP® DocuPrint NPS

With DocuPrint NPS, you must specify duplex as a job attribute through a submission option or a virtual printer, and in the VIPP® file using DUPLEX\_on.

## Using SETMEDIA with DocuPrint NPS

To obtain VIPP® media calls working on DocuPrint NPS, specify media in these areas:

- In the DSC statements. These must be specified in the submission file, not in the JDT. However, it is possible to avoid using the DSC statements for this purpose when you create a virtual printer that includes media specifications or when specifying them in the submission command line:

```
lpr -C" (media=a4::white, media=a4::red, ...) ."
```

- On DocuPrint NPS using the Set Tray command.
- In the print job using the SETMEDIA command.

## Example

```
%!
%%Title: ex8.nm
% Commands: SETMEDIA
%
% DSC definitions so DocuPrint NPS can make this job
% ineligible if the media required is unavailable
% 8.5 x 11 = 612 x 792
% A4 = 595 x 841
%
% When using SETMEDIA in LINE MODE....
% DSC comments relating to DocumentMedia must be in the
% SUBMISSION file (the data file or a file that calls
% the data file & .jdt/.dbm)
% DSC comments will have no effect if they are in the JDT or DBM.
% NOTE: SETMEDIA relies on the 'type' NOT on 'tag' and not 'color'
% SETMEDIA can be used with three parameters <type> <color> and <weight>.
% For this example to work:
% One tray on DocuPrint NPS must be set to A4:YELLOW:white and
% one tray on DocuPrint NPS must be set to A4:GREEN:white and
%
% Key: <Tag Name> <width> <height> <weight> <color> <type>
%%DocumentMedia: dummy 595 841 0 white ()
%%+ YELLOW 595 841 0 white YELLOW
%%+ GREEN 595 841 0 white GREEN
PORT

1250 3200 MOVETO
/NHEB 14 SETFONT
50 SETLSP
(This is an example of using SETFORM and SETMEDIA - default media) SHC
PAGEBRK

% Change media
(YELLOW) SETMEDIA
1250 3200 MOVETO
(This is an example of using SETFORM and SETMEDIA - YELLOW media) SHC
PAGEBRK

%Change media
(GREEN) SETMEDIA
1250 3200 MOVETO
(This is an example of using SETFORM AND SETMEDIA - Greenmedia) SHC
PAGEBRK
```

## VIC CONSIDERATIONS FOR FREEFLOW PRINT SERVER

This section provides information specific to the FreeFlow Print Server controller, it includes:

- [Production and Demonstration mode](#)
- [Preprocessing capabilities](#)
- [FreeFlow Print Server security settings and FTP/ NFS](#)
- [FreeFlow Print Server Watched Folders](#)
- [FFPS and PPR configuration](#)
- [VI Project Container Filter and FreeFlow Print Server](#)

### Production and Demonstration mode

VI Compose delivered on FreeFlow Print Server devices is enabled for VIPP® Demonstration mode. This mode allows you to use the full set of VIPP® features on the FreeFlow Print Server. Documents larger than 200 pages abort at page 200 and the Selected pages: 1 200 message displays Refer to [FreeFlow VI Compose Licensing and Installation](#) for more information.

To use VIPP® for production printing, you must activate Production mode by purchasing and installing a device-specific VIPP® license string on the FreeFlow Print Server controller. For installation instructions, refer to [VI Compose Installation on FreeFlow Print Server \(Solaris\) DFEs](#).

### Preprocessing capabilities

Printer preprocessing capabilities have been added to the FreeFlow Print Server by Decomposition Services. Refer to [Decomposition services](#) for more information.

### FreeFlow Print Server security settings and FTP/ NFS

The FreeFlow Print Server documentation for information on the FreeFlow Print Server security settings required to enable FTP and NFS. Security settings are specific to the FreeFlow Print Server software level installed on the print device.

### FreeFlow Print Server Watched Folders

The FreeFlow Print Server has introduced the concept of a Watched Folder for job submission to a queue. This can make job submission as simple as drag and drop. For example you could create a Watched Folder and mount it to the PC. The Watched Folder could be configured to submit a job to a queue named VIPP®\_JOB. The queue VIPP®\_JOB could be configured to use the VI Project Container Filter option (see below). This would enable the user (or application) on the PC to simply drag and drop a .vpc file into the FreeFlow Print Server Watched folder. The Watched folder would submit the job to the VIPP®\_JOB queue for processing. The VPCF filter option would be able to deploy the VI files and print the job submission file.

### FFPS and PPR configuration

On FFPS, when a VIPP® job is sent to a parallel RIP workflow (PPR), it is recommended that you normalize the VIPP® job, create a .vpn file before printing, or as part of the job printing process. For duplex jobs it is critical to set the PPR chunk size to an even number to avoid splitting sheets between front and back and printing extra blank sheets.

### VI Project Container Filter and FreeFlow Print Server

The VI Project Container Filter (VPCF) is a job filtering mechanism that can be applied to one or more queues on a FreeFlow Print Server system. When the VPC Filter is set up on a queue, it processes all VIPP® jobs sent to

the queue via any print path gateway. Non-VPC files are passed through the queue unfiltered and VPC files are filtered / expanded, resources deployed, etc. according to the filter's setup configuration for the given queue. The VPCF GUI enables users to specify how the filter will process VPC files and manage their resources. VPCF software is installed as part of the FreeFlow Print Server software installation. Once VPCF has been installed, VPCF updates from a FreeFlow Print Server installation will retain all existing VPCF configuration files.

## Deleting all VPCF Configuration Files

To delete all VPCF configuration files, before you install the FreeFlow Print Server, do the following steps:

1. Login as root
2. Type `pkgrm XRXvpcf`



Note: After you delete all VPCF configuration files, remove the VPC Filter from all filter-enabled queues. Refer to the topic *Removing a VPC Filter from a FreeFlow Print Server Queue*.

## Setting up the VPC Filter on a FreeFlow Print Server Queue

Follow these steps to set up VPCF:

1. Select the **FreeFlow Print Server Queue Manager** and bring up the properties page of a new or existing queue.
2. Select **Input/Format** and set the format to **PostScript**. Set any other input options required for this queue.



Note: When filters are applied, never select the Input Streaming option. If you select the option, jobs do not process correctly.

3. Select **Job Filter**, enable the **Apply Filter** check box, and select **VI Project Container Filter (VPCF)** from the list of filter names.



Note: This action indicates to the FreeFlow Print Server that the VPC filter is applied to all VPC jobs sent to the queue. Job filter options are available only for multiple-queue systems.

4. To set up a new queue, select **Add Queue** followed by **Close**, then re-open the newly created queue.
5. To launch the VPCF GUI, select **Setup**.



Note: The GUI window is launched from a terminal and can take several seconds to appear.

## Using the VPC Filter GUI

When invoked from FreeFlow Print Server, VPCF is opened into a separate GUI window.

The VPCF GUI contains a drop-down list box titled VPC Processing Options. These options are:

Print and Forget (default)	This option prints the job(s) and removes any downloaded resource files from the printer upon job completion. There are no deploy options for Print and Forget.
Deploy and Print	This option deploys and saves the resource files on the printer and prints the jobs. Deploy and Print options are: Replace existing resources (default). When selected, resources downloaded to the existing printer will overwrite resources of the same name that resources already exist on the printer. When this option is not selected, overwriting will not occur. Keep print file deleting the

print file is the default. This is no longer an option in VPCF 4.0 and subsequent versions. The print file is always stored on the printer when either of the deploy options is selected.

Deploy Only	Deploys and saves the resource files on the printer. Deploy options are: Replace existing resources (default). When selected, resources downloaded to the existing printer overwrite resources of the same name that resources already exist on the printer. When this option is not selected, overwriting will not occur. Keep print file, deleting the print file is the default. When selected, the print file is kept on the printer. When this option is not selected, all files except the sample print file are deleted.
Print the list of deployed resources	(default) When selected, a list of deployed resources and their status is printed. Loaded, A new resource file from the VPC file was saved to the printer. Replaced, the existing resource file was replaced with the resource from the VPC file. Not replaced, the existing resource file was not replaced with the resource in the VPC file

When this option is not selected, a summary of the total count of resources deployed is printed.



Note: When either of the deploy options is selected, the VPC Filter creates a 4-byte file called `semaphore` in the VPC project local folder. The `semaphore` file is left in the folder intentionally. After deployment, you can remove the file from the folder.

Select the **Delete all obsolete VPCF configuration files** checkbox to delete all VPCF configuration files that are associated with print queues that no longer exist on the system. Selecting this checkbox will not delete VPCF configuration files that are associated with existing print queues that no longer have the VPC Filter applied.

The GUI has three additional buttons:

OK	currently selected values are saved and the GUI will close.
Reset	GUI options are reset to their last saved values and the GUI will not close.
Cancel	The filter's queue settings remain unchanged and the GUI will close.

### Removing a VPC Filter from a FreeFlow Print Server Queue

1. Select the **FreeFlow Print Server Queue Manager** and bring up the properties page of a filter- enabled queue.
2. Select the **Job Filter** option, and uncheck the **Apply Filter** checkbox.
3. Select **OK**.

### VPC Submission Files

When a VPC file is submitted to a VPC filter-enabled queue, all production type files that are contained in a VPC file are printed.

A sample type file in a VPC file is printed only when there are no production files contained in the VPC file.

VPC files that contain production jobs that use VIPP® Demographics or Normalization features are not supported.

It is possible to create and submit a VPC file that contains multiple production files and job tickets, however, FreeFlow Print Server can apply only one job ticket per VPC file or job ID. The job ticket of the first job that is printed from the VPC file is applied to all succeeding jobs within the VPC file.

## VPC Error Codes

- 129 VPCF\_EC\_FILTER\_NOT\_INITIALIZED\*: The General Filter Mechanism failed to call the **VPCF** filter initialization routine.
- 130 VPCF\_EC\_TV\_FAILURE\*: One of the Tag Value (TV) library routines used to delete files upon job completion has failed.
- 131 VPCF\_EC\_PROJECT\_CREATION\_FAILED: The **VPSDK** failed to create an empty project for the submitted VPC file. Verify that the **SETPATH** paths in `/usr/xgf/src/xgfunix.run` are correct and that the directories in those paths exist.
- 132 VPCF\_EC\_OPEN\_ARCHIVE\_FAILED\*: The **VPSDK** failed to extract the files that are contained in the submitted VPC file.
- 133 VPCF\_EC\_PRINT\_FILES\_NOT\_FOUND: The submitted VPC file does not contain any print files.
- 134 VPCF\_EC\_RESOURCES\_DEPLOYED\_PRINT\_FILES\_NOT\_FOUND: The resources that are contained in the submitted VPC file deployed successfully, but a request to print failed because the VPC file does not contain any print files.
- 135 VPCF\_EC\_ITERATOR\_CREATION\_FAILED: The **VPSDK** failed to create an iterator for the submitted VPC file. Verify that the **SETPATH** paths in `/usr/xgf/src/xgfunix.run` are correct and that the directories in those paths exist.
- 136 VPCF\_EC\_FILE\_OPEN\_ERROR\*: An attempt to open a file failed.
- 137 VPCF\_EC\_OPEN\_PROJECT\_FAILED\*: The **VPSDK** failed to open a project for the submitted VPC file.
- 138 VPCF\_EC\_CONFIG\_ACCESS\_FAILED\*: An attempt to access the VPC Filter's configuration file failed.
- 139 VPCF\_EC\_INVALID\_PROJECT\_TYPE: The submitted VPC file contains an invalid PDL type. Valid PDL types are: VIPP®, PDF, PS, and PPML-GA.
- 140 VPCF\_EC\_NO\_PROJECT\_TYPE\_AVAILABLE\*: No PDL type is specified for the submitted VPC file.
- 141 VPCF\_EC\_OVERWRITING\_OF\_PROJECT\_NOT\_ALLOWED: The Print and Forget VPC processing option cannot be used to process the submitted non-VIPP® VPC file. The project contained in the VPC file exists already on the system, and the Print and Forget option does not allow the overwriting of files. You can use the Deploy and Print and Deploy Only VPC processing options to process the VPC file, as long as the Replace Resources option is `TRUE`.
- 142 VPCF\_EC\_FILE\_DATA\_MISSING\_OR\_CORRUPTED\*: An attempt to read data from a file failed. The data is missing or corrupted.
- 143 VPCF\_EC\_UNABLE\_TO\_LOCATE\_VPF\_FILE\*: An attempt to locate the VPF file failed.



Note:

### VPSDK

The FreeFlow VI Projects Software Development Kit.

### System errors contain an asterisk

All errors that contain an asterisk (\*) are system errors. Contact a Xerox Support Representative immediately and report the system error.

## VIC CONSIDERATIONS FOR DOCUCOLOR DEVICES

Special consideration must be taken when coding VIPP® jobs or VI Projects for the DocuColor printer series. The DFEs that power the DocuColor family of printing systems do not support the full set of VIPP® commands as described in the *VIPP® Language Reference Manual*. Information about these considerations is included in the following topics:

- **Unsupported commands**
- **Required VIPP commands**

### Unsupported commands

This section lists the VIPP® commands that are not supported on DocuColor printing systems that use Scitex or EFI DFEs. Use of these commands can cause the VIPP® job to abort.

- **STARTOFSET**

The following commands have limited support. For more information, contact the DFE Vendor.

- **DUPLEX\_on** and **DUPLEX\_off**
- **TUMBLEDUPLEX\_on** and **TUMBLEDUPLEX\_off**
- **SETMEDIA** and **SETMEDIAT**

### Required VIPP® commands

The following VIPP® commands are required for VIPP® jobs or VI Projects to work on DocuColor printing systems that use Scitex or EFI DFEs. Without the following commands, the job can abort.

- **SETPAGESIZE**
- Use the **SETPAGESIZE** command with one of the following orientation commands:
  - **PORT**
  - **IPOINT**
  - **LAND**
  - **ILAND**

## VIC CONSIDERATIONS FOR THE OFFICE ENVIRONMENT

Due to the implementation of printing VIPP® jobs or VI Projects on Xerox office printers, all resources are read in memory as procedures. PostScript files with embedded images using currentfile do not work when defined as forms or segments.

There are several ways in which to work around this problem; each is described in these sections:

- [Using Decomposition Services](#)
- [Using CACHE SCALL and CACHE SETFORM](#)
- [Running the file in a form definition](#)
- [Running the file in an in-line form definition](#)

### Using Decomposition Services

PostScript forms can be preprocessed and referenced as described in [Decomposition services](#). This solution is the most suitable from a performance aspect. Use this capability when a stackoverflow error occurs due to the processing and printing of large forms.

### Using CACHE SCALL and CACHE SETFORM

Use CACHE SCALL or CACHE SETFORM to place any TIFF, JPEG, PostScript, or EPS element into the job. The CACHE command provides a single command from which all these formats can be accessed. CACHE SETFORM allows the use of a TIFF file as a form, and also removes the limitation related to large PostScript or EPS files and PostScript or EPS files with embedded images.

### Running the file in a form definition

This section provides an example of running the file in a form definition.

Example:

```
%!
%%Creator: CAS
%%DocumentName: form1.frm
{
  (path to the file) run
} FSHOW

%!
%%Creator: CAS
%%DocumentName: list1.jdt
....
(form1.frm) SETFORM
....
```

### Running the file in an in-line form definition

This section contains an example of using SETFORM when a PostScript file with embedded images using currentfile is defined as a form or a segment. This solution works only on systems that include a hard disk.

Example:

```
%!
%%Creator: CAS
%%DocumentName: list1.jdt
....
{ (path to the file) run } 0 SETFORM
....
```

## STORING VIPP® RESOURCES ON OFFICE PRINTERS

You can store VIPP® resources on a local disk on the target device. To load resources to desktop or office printers, and to send self-contained VIPP® jobs or VI Projects to diskless printers, use the VIPP® Manage utility that is

provided with VI Compose. You can use a third-party tool to load VI Compose and VIPP® resources to a device of this type. For more information, contact a Xerox representative.


## VIPP® COMMAND DIFFERENCES

VIPP® commands can differ across printer families. This section addresses VIPP® command differences across the DocuPrint 65 printer, FreeFlow Print Server, and DocuPrint NPS printer families.

Differences in these commands are summarized in the following table.

The use of TIFF files in VIPP® jobs is discussed in relation to each printer family.

VIPP® COMMANDS	DP/DT 75 AND 90	FREEFLOW PRINT SERVER	DOCUPRINT NPS
<b>% %DocumentMedia</b>	Not required	Not required	Required when the <b>SETMEDIA</b> or the <b>SETMEDIAT</b> command is used
/Bind option <b>SETFINISHING</b>	Not applicable; ignored	FreeFlow Print Server	Not applicable; ignored
<b>DUPLEX_off</b> <b>TUMBLEDEX_off</b>	Speed improves with simplex and duplex because blank pages are not imaged on the back sides of simplex pages.	Speed improves with simplex and duplex because blank pages are not imaged on the back sides of simplex pages.	Default setting in the off state. When a job is submitted with the duplex option, it provides simulated mixed plex. No speed difference with mixed simplex and duplex jobs unless / MixPlexCount is set using <b>SETPARAMS</b> .
/Offset option <b>SETFINISHING</b> with <b>ENDOFSET</b> and <b>ENDOFRUN</b>	Cannot offset within a set. Cannot offset on page boundaries, only on set boundaries.	Cannot offset within a set. <b>ENDOFSET</b> and <b>ENDOFRUN</b> are the same.	To use, enable the MultiSet feature. Can offset on page boundaries.
/Staple option <b>SETFINISHING</b>	DualPortrait is supported.	SinglePortrait, SingleLandscape, and DualLandscape are supported.	Not applicable; ignored
/Staple option <b>SETFINISHING</b> with <b>ENDOFSET</b> and <b>ENDOFRUN</b>	This command addresses internal stapling only. External Finisher uses <b>SETOBIN</b> , <b>ENDOFSET</b> , and <b>ENDOFRUN</b> are the same. Use the FreeFlow Print Server.	This command addresses internal stapling only. External Finisher uses <b>SETOBIN</b> , <b>ENDOFSET</b> , and <b>ENDOFRUN</b> are the same. Use the FreeFlow Print Server.	Enable the MultiSet feature. <b>STAPLE_on</b> must be coded in the JDT or the data file. System must have the switching module. Staple attribute set with each job.

VIPP® COMMANDS	DP/DT 75 AND 90	FREEFLOW PRINT SERVER	DOCUPRINT NPS
<b>STARTOFSET</b>	Command acts as a set delimiter. Place the command at the beginning of the first page of a set in the submitted data file.	Command acts as a set delimiter. Place the command at the beginning of the first page of a set in the submitted data file.	Not applicable; unpredictable results
 Note: You can use the <b>STARTBOOKLET</b> and <b>ENDBOOKLET</b> commands across all platforms. The <b>STARTBOOKLET</b> and <b>ENDBOOKLET</b> commands are preferable to the <b>STARTOFSET</b> , <b>ENDOFSET</b> , and <b>ENDOFRUN</b> commands.			

## VI Compose and FreeFlow Makeready

This section describes the interface between VI Compose and FreeFlow Makeready. It also applies to the interface between VI Compose and FreeFlow Makeready, use this information when using FreeFlow Makeready. In the following sections of this document the FreeFlow Makeready module is referred to as Makeready.



Note: Makeready is required along with the VI Compose/FreeFlow Makeready API to enable the VIPP®/RDO interface. VI Compose can process only TIFF-based RDO files.

Installation and configuration of Makeready is described in the FreeFlow Installation Guide. Installation of VI Compose/Makeready API is described in [VI Compose / Makeready API Installation](#).

### VI COMPOSE/MAKEREADY API OVERVIEW



Note: This feature is not available in line mode VIPP® for the JDT (Job Description Ticket).

The VI Compose/Makeready API enables VIPP® to access Makeready Raster Document Object (RDO or .rdo) files as VIPP® resources from within a VIPP® DBM or native mode file that can use RDO files as VIPP® resources. The RDO files are used much like VIPP® uses images or forms stored in its own subdirectories on the printer's hard disk.

The VI Compose/Makeready API, which is transparent to the user, is enabled when the Makeready software is installed. Once installed, Makeready will create a VDI file with a .vdi extension in addition to the CON (TIFF) folder and RDO file currently generated every time a document is created or edited in Makeready.



Note: You do not have to install VI Compose on the Makeready PC. Instead, install the VI Compose/Makeready API that is in the Makeready folder on the VI Compose CD.

### VDI file

The VDI file is created in the parent directory of the directory containing the RDO files and corresponding TIFF directories. The VDI is for VIPP® internal use only.



Note: Do not edit the VDI file. The VDI file is referenced when a VIPP® **RUN** command is coded in a DBM or native mode file. Although VIPP® uses the VDI file, the usage is transparent to the Makeready user who edits the RDO file. When you make changes to RDO files, always use Makeready. The Makeready interface enables the creation of the .vdi file that VIPP® uses as a map to the stored images. The stored images are printed using VIPP®.

You must share the directory that contains the VDI files. This shared VDI directory must contain a subdirectory that contains the RDO files and CON (TIFF) directories. Because this setup relies on NFS mounted systems, a UNIX system administrator should assist in setting up the NFS mount and export rights. Only the VDI directory is NFS mounted, the rest of the Makeready repository can be isolated from the network.

### RDO files/VIPP® Text Object

There is a field in the RDO structure known as the VIPP Text Object. This field appears as a menu item in the Makeready module. It allows selection of an RDO file, selection of a document boundary within the file, (chapters, start or end pages, and so on, and type-in text. The intent is to type VIPP® layout commands into the text object that invoke VIPP® formatting instructions.



Note: Use the Makeready module to create, edit, move, or delete the RDO document. When you fail to use this interface, the VIPP®-required VDI file is not created. VI Compose ignores any job ticket information that is saved with the RDO document. To control page formatting, use VIPP®. When VIPP® Text is inserted on a page, the text is required as the first type of object on that page.

Even though this field will not appear on every Makeready system at start-up, installation of VI Compose/ Makeready API, part of the VI Compose CD is required.

### **RDO files/Digital Library and VIPP® Cabinet**

The Digital Library for Makeready will be modified to include a VIPP® Cabinet that can be used to manage all the RDO documents accessed by VIPP®, in addition to being the repository for VIPP® files. This cabinet is set up for your convenience, it does not have to be used. Place the Makeready files in any directory accessible to the VIPP® directory via NFS mount.

### **HOW VI COMPOSE CALLS THE RDO FILE**

The VIPP® data file is submitted to the printer just like in any normal VIPP® job. Keep these points in mind:

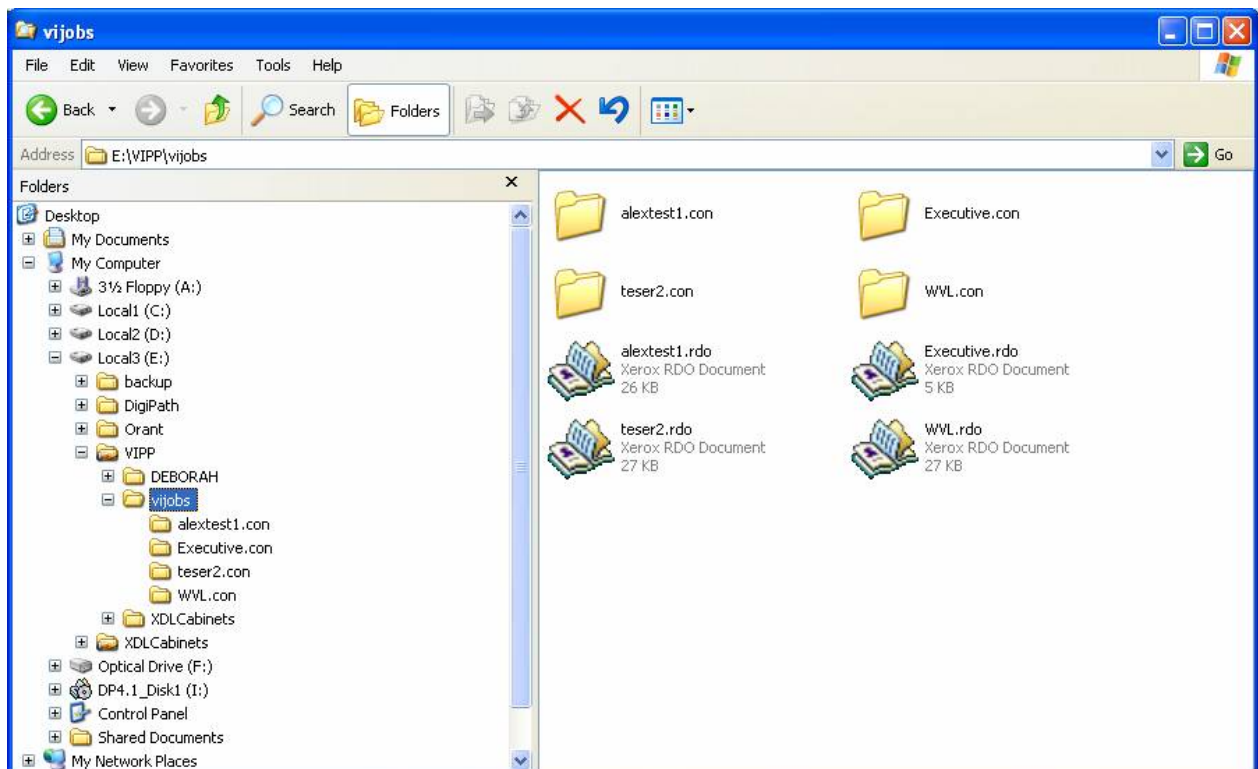
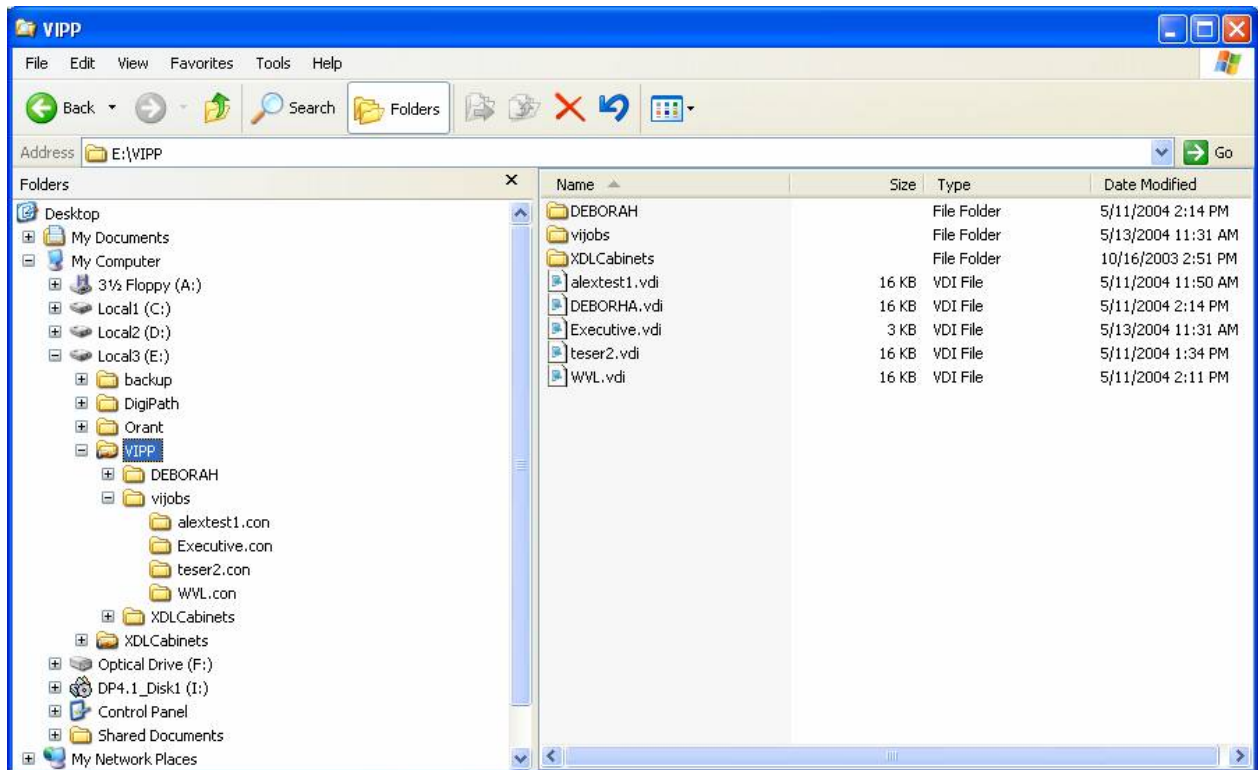
- The DBM or native mode file can contain calls to a RDO document.
- The VDI directory that contains the RDO directory structure must be NFS mounted to the print device. This is the directory structure used by the Makeready module when the RDO was created or edited.
- The VIPP® formatting instructions can be in the DBM or in the RDO file parsed from the VIPP® Text Objects. This allows the submission of the database extraction as input variable data that calls a DBM file or a VIPP® native mode file.
- The DBM may select one or several documents from the Makeready Digital Library the VDI directory and format them based on the variable data content.
- The VIPP® RUN command has been modified to use the VDI directory and understands the directory structure within the VDI directory, when the correct mount point has been established.
- VI Compose cannot print an RDO document sent directly from the Makeready system. The RDO file must be called by way of a VDI file in a VIPP® DBM or native mode job.

When an RDO document is created or modified using Makeready, an RDO file and a CON directory are created, which contain the TIFF images referenced in the RDO file. Makeready will also create a VDI file one level above the directory that contains the RDO file and CON directory. Although the RDO file is used when making modifications to the document from within Makeready, the VDI file is referenced when coding the call in the VIPP® program. For example, an RDO file called mytest.rdo in the VDI directory would be called out in a VIPP® program using this syntax:

```
(MYTEST.vdi) RUN
```



Note: The VDI file is generated in the VDI directory, not in the document directory where the RDO files are located.



## VIPP® TEXT

VIPP® Text is optional. You do not have to embed it into RDO documents to have VDI files created.

VIPP® Text allows the creator of the RDO file to add hidden VIPP® layout commands to the file. The commands are executed as part of a normal VIPP® job. For example, the variable name and address label can be embedded in an RDO document that is encapsulated in an in-line **SETFORM** definition. When the document is called as a resource to a VIPP® job, the VIPP® Text or VIPP® commands that are contained in the VIPP® Text label are executed as a normal part of the VIPP® job. The variable labels in the RDO document are replaced with the information from the customer data base file.

Another use of VIPP® Text is to emulate the Makeready job ticket information that is ignored by VIPP®. For example, VIPP® Text can be used to place VIPP® media calls to replace media information in the Makeready job ticket.

This feature is not intended for complete VIPP® programs or individual page-marking commands that are not embedded within a **SETFORM** procedure, such a **SH** or **SHP**. This feature is designed for VIPP® layout commands such as **SETMEDIA**, **SETFORM**, **SETPAGEDEF**, **SETMULTIUP**, and **DUPLEX\_on**.

You can use Makeready to add Multiple VIPP® Text lines to an RDO document. The VIPP® Text line of VIPP® commands requires a beginning %%VIPP® prefix. The commands are executed as part of a normal VIPP® job. Page-marking commands can be encapsulated within an in-line form definition.

When the commands or syntax that are used in the VIPP® Text are invalid, an error occurs, and the job aborts at the print device during the print run. No syntax checking is performed within the RDO file, so care must be taken to ensure that no errors are made when entering the VIPP® commands. When the data field that is referenced in the VIPP® Text file is invalid or unavailable, the job aborts at run time.

Use this feature with caution and consider it as a source for potential problems when troubleshooting VIPP® jobs using RDO files and VIPP® Text. Always keep these points in mind when using this feature:

- VIPP® Text is not designed to handle complex VIPP® code.
- Each VIPP® Text entry must be made on a single line.
- The maximum line length is 132 characters, including spaces.
- The VIPP® Text string requires a beginning %% VIPP® prefix, without spaces.
- Enter the VIPP® commands on a single line and use the %% VIPP® prefix.
- You can enter multiple VIPP® commands on a single line, if you separate the commands with a space character.
- No VIPP® comments ( %) are allowed in a VIPP® Text entry.
- When VIPP® Text is inserted on a page, place the VIPP® Text before any other type of object on that page.

When you assign multiple VIPP® Text fields to a single page, the fields are processed in the order in which they were assigned. Do not use the VIPP® Text fields for programming. Instead, to create embedded procedures, use the **SETVAR** or **XGFRESDEF** command in the VIPP® code, then invoke the procedures from within the VIPP® Text.

## Entering VIPP® Text

To create an RDO object with VIPP® Text:

1. Select an existing RDO file or create one using Makeready.
2. Open the file. When this is a multi-page file, select the page on which the VIPP® Text will be placed.
3. From the Menu bar, select **Insert > VIPP Text**.
4. Move the VIPP® Text object to a position before any other type of object on the page.

A window will be displayed. You are prompted to enter a text label for the VIPP® Text object. The VIPP® Text label is user-defined.



Note: The syntax for the VIPP® Text requires valid VIPP® code. No syntax checking is performed. When an error occurs, the error aborts the VIPP® job during the print run.

Pressing **Enter** ends the VIPP® Text dialog. Once you press **Enter**, the VIPP® Text entry is complete. Following is an example of a VIPP® Text entry:

```
%%VIPP { 300 300 MOVETO ($$Fname. $$Lname.) VSUB SH } SETFORM
```

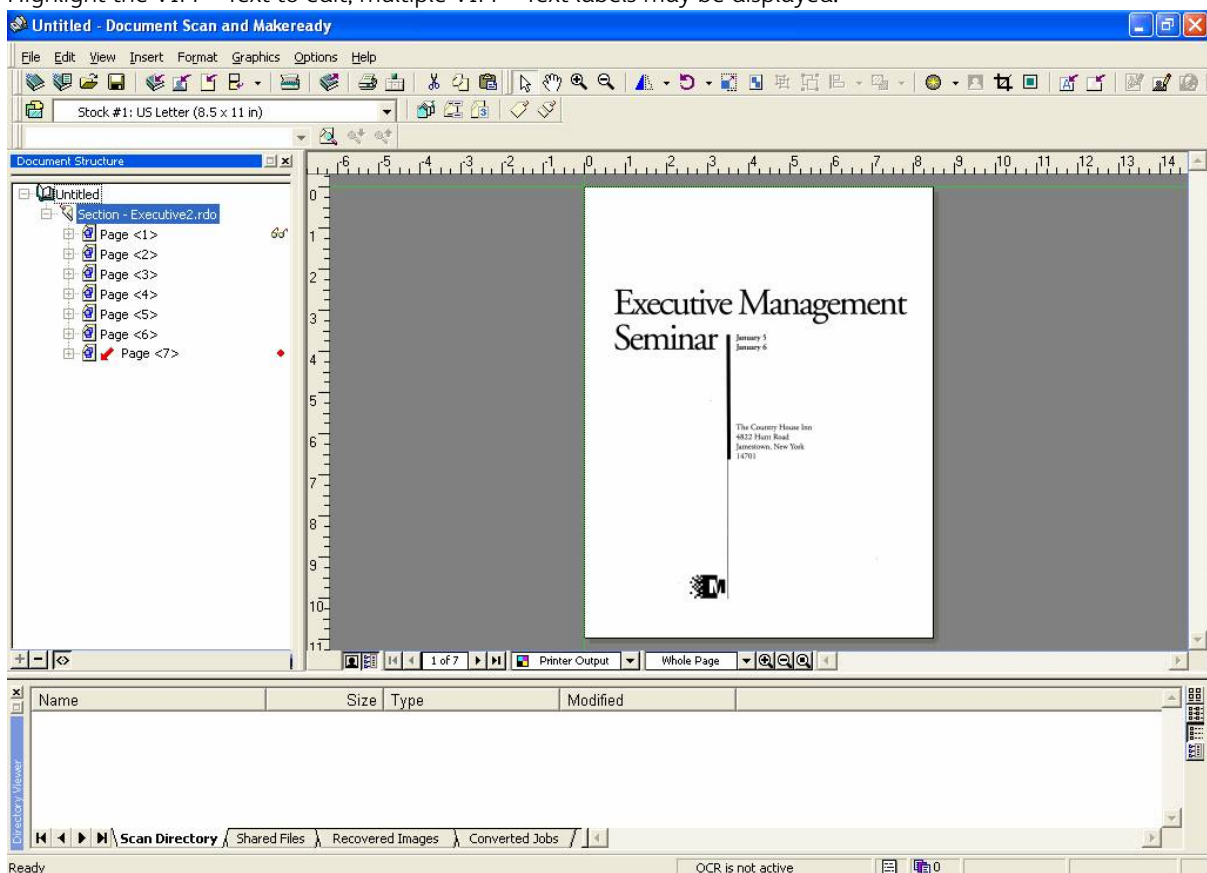
**Where:**

Fname and Lname are fields in a database record.

### Editing a VIPP® Text Entry

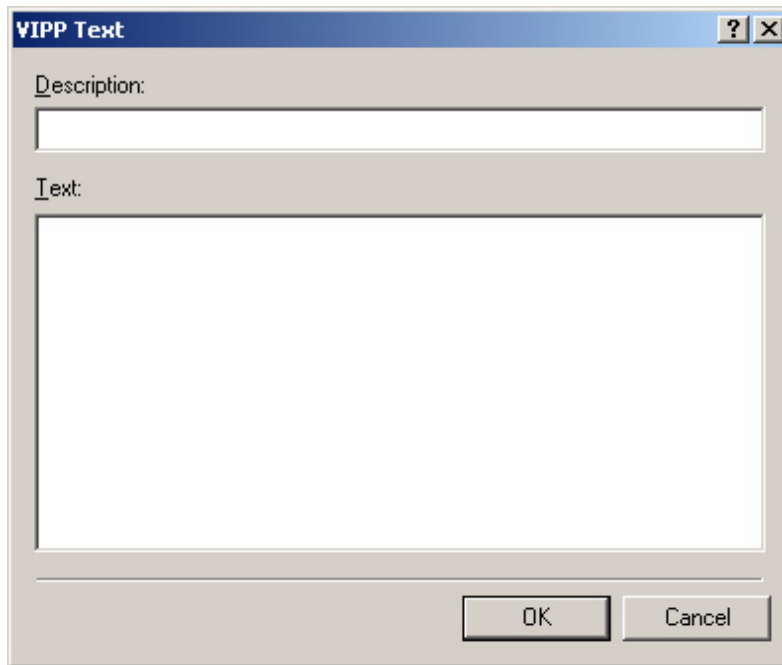
To edit an existing VIPP® Text entry:

1. Select the RDO document with the VIPP® Text entry.
2. Highlight the VIPP® Text to edit; multiple VIPP® Text labels may be displayed.

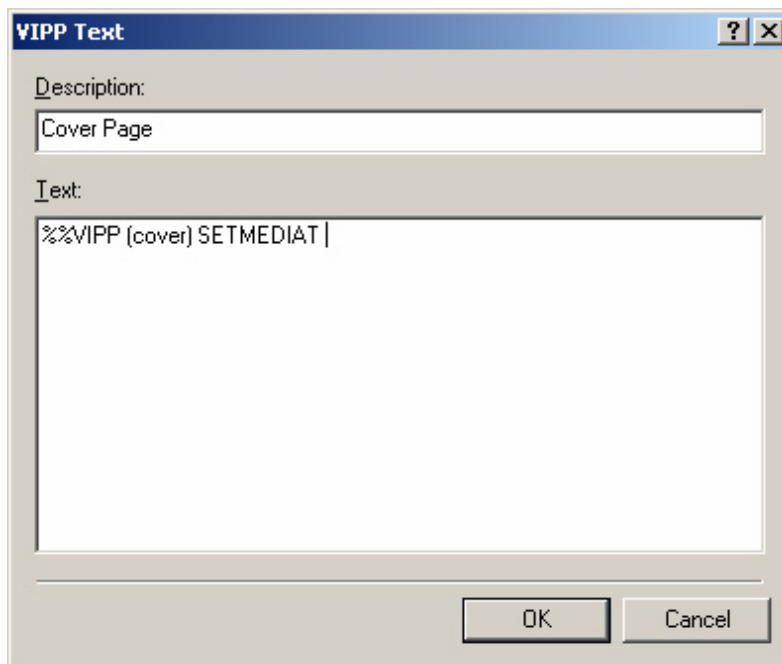


3. Right-click the highlighted text and select the **Specialty Imaging Properties** option.

The VIPP® Text interface is displayed allowing you to edit the contents.



The screenshot shows the 'VIPP Text' dialog box with a blue title bar containing a question mark icon and a close button. The dialog has two main sections: 'Description:' with a single-line text input field, and 'Text:' with a large multi-line text input area. At the bottom right, there are 'OK' and 'Cancel' buttons.



The screenshot shows the 'VIPP Text' dialog box with the same layout as the previous one, but with content entered. The 'Description:' field contains the text 'Cover Page'. The 'Text:' field contains the text '%%VIPP (cover) SETMEDIAT |'. The 'OK' and 'Cancel' buttons are at the bottom right.

### USING AN RDO FILE IN A VIPP® JOB

To use an RDO file in a VIPP® job, enable the VI Compose Makeready API. The VDI directory must be NFS-mounted on the target print device and have the correct attributes to allow the print device to access files in the VDI directory.

The `xgfunix.run` or `xgfdos.run` file must be modified on the printer to include a path to the correct directory within the Makeready VDI directory.

To place variable data on various pages of the document, use the **ENDPAGE** command, including the incrementation of a page count variable and a **CASE** statement that refers to it. Refer to the examples in the **RUNDD** and **RUNTIF** command descriptions in the *VIPP® Language Reference Manual*.

Use the correct VIPP® syntax to call out an RDO file:

```
(filename.vdi) RUN
```



Note: VIPP® file names are case-sensitive. The file name that is enclosed in parentheses, including the file name extension, are required to match the case of the file name on the Makeready cabinet.

## LCDS Migration

This section contains information, examples, hints, and tips for LCDS-to-VIPP® migration. This section includes the following information:

- [General Description](#)
- [Basic LCDS](#)
- [LCDS and VIPP resources](#)
- [LCDS and VIPP commands](#)
- [Conversion examples](#)
- [LCDS migration hints and tips](#)

### GENERAL DESCRIPTION

VI Compose uses the same Dynamic Document Construction concepts as the Xerox LCDS printers. Fonts, forms, and images reside on the printer and formatting is controlled by a template file (JDT) called by the job. The resources are merged with the data at the printer. The result is an efficient printing model that saves valuable network bandwidth. With VIPP®, only the variable data is sent to the printer.

A VIPP® programmer can take advantage of the built-in PCC tables, (/ANSI, /IBM1403, /IBM1403A), page and record conditional processing, and the PROCESSDJE command to create a VIPP® job that can emulate or enhance the original LCDS job without requiring changes to the original data set. These features include rounded corners, data driven graphics, reusable data elements, and multiple form layouts. VIPP® can also be used to add highlight or full color to LCDS jobs converted to VIPP® and sent to a highlight or full color print device or PDF output.

An LCDS data stream can be migrated to VIPP® when the data stream does not contain these:

- Metacode object
- In-line objects such as resource files, fonts, forms, and graphics downloaded in the data

The LCDS data stream can contain Dynamic Job Descriptor Entry (DJE) records. The VIPP® programmer must code a JDT file that captures information from the DJE record and substitutes it with a VIPP® command or procedure.

When considering LCDS-to-VIPP® migration, you must:

- Obtain copies of original resources, which must be converted to PostScript resources. This is an opportunity to update old resources and add new features supported by the PostScript print device.
- Review the original job. VIPP® cannot be used to create the new job when the job contains in-line resources or metacode.
- Review the JSL and data file. At this point, decide whether to emulate the LCDS job, recode the job to meet new requirements, or use new functionality provided by VIPP®. In some cases it may be easier to recode the job than to recreate the original job.
- Convert existing resources or create new resources. Several third-party vendors provide the services to convert LCDS resources.
- Code the new application in the VIPP® language. Xerox provides services that can assist in VIPP® programming. For further information, contact a local Xerox representative.

## BASIC LCDS

LCDS jobs are typically comprised of a data stream, job resources, and control files such as Job Source Libraries (JSL), Copy Modification Entries (CME), Page Descriptor Entries (PDE), and Dynamic Job Descriptor Entries (DJDE).

The table below reviews some of the LCDS commands and parameters used to control print format and processing. These must be emulated with VIPP® commands when migrating a job to VIPP®.

LCDS COMMAND / PARAMETER	FUNCTION
CME	Applies font or ink changes within variable data for certain parts of the report output, possibly replacing them with predefined static data. CMEs may vary from one copy to another.
PDE	Controls page format orientation and fonts.
DJDE	Sent through input data stream, specifies the forms, fonts, and graphics to print or sets up various dynamic changes in the layout.
RTEXT	Specifies the text to be printed on a separate page preceding a report.
RPAGE	Specifies the position of the current, or next, logical page on the physical sheet.

## LCDS AND VIPP® RESOURCES

To convert an LCDS job to VIPP®, all LCDS control files and commands must be recoded as VIPP® equivalents. This requires the creation of a VIPP® JDT for use in place of the JSL, CME, PDE, RTEXT, and so forth.

A JDT is made up of VIPP® commands and functions that format and control the data on a page. The JDT sets the orientation, margins, fonts, forms, and so forth.

### Creating the JDT file from the JSL

There is no straightforward way to create a JDT from a LCDS based JSL. Each JSL command can be converted to one or several VIPP® commands. Typically, the VIPP® programmer codes this manually using VI Design Pro, or a third party design tool.

LCDS resource files must be converted from their native source to a PostScript resource. Many of the third-party tools available can be used to convert LCDS resources to VIPP®.

The table below lists some of the LCDS resources and their PostScript or VIPP® equivalent.

LCDS RESOURCE	POSTSCRIPT RESOURCE
.JSL	Job Descriptor Ticket (.JDT)
.FNT	PostScript Type I or III fonts
.FRM	PostScript print file or Native VIPP® form
.IMG	TIFF or JPEG images
.LGO	VIPP® segments or TIFF/JPEG files

## LCDS AND VIPP® COMMANDS

Many of the functions initiated by the LCDS commands can be converted to VIPP® by using VIPP®-equivalent commands.

The following table lists some common LCDS commands and their VIPP® equivalents. The commands are not always one-to-one relationships. In some cases, several VIPP® commands are necessary to provide equivalent functionality.

LCDS COMMANDS	VIPP® COMMANDS
CODE	SETENCODING
RECORD LENGTH and STRUCTURE	SETBUFSIZE
TABLE and CRITERIA	SETRCD, SETPCD
RSELECT and RDELETE	SETRCD and Conditional RPEs
ROFFSET and OFFSET	SETFINISHING, CPCOUNT, ENDOFSET, STARTOFSET, BEGINPAGE
BANNER	SETPCD, SETJDT, BEGINPAGE
RSTACK	SETSKIP, SETPCD, SETJDT, BEGINPAGE
RAUX	SETPCD, SETMEDIA, BEGINPAGE
RPAGE and SIDE	SETPCD, SETSKIP, NEWFRONT, NEWSIDE, NEWBACK, BEGINPAGE
FORMS and BFORM	SETFORM, SETBFORM
COPIES	SETCYCLECOPY
COLLATE	COLLATE_on, COLLATE_off
COVER	SLIPSHEET
CYCLEFORMS	[ ... ] SETFORM
DESTINATION	SETOBIN
DUPLEX	DUPLEX_on, DUPLEX_off, TUMBLEDUPLEX_on, TUMBLEDUPLEX_off
FEED and STOCKS	SETMEDIA, SETMEDIAT
PMODE	PORT, LAND, IPORT, ILAND
BEGIN	SETMARGIN, DJDEBEGIN
Multiple BEGINs	SETMULTIUP, DJDEBEGIN
FONTs	SETFONT, INDEXFONT, RPEDEF
GRAPHIC and IMAGE	MOVETO, ICALL, SCALL
MODIFY and CME	BEGINRPE, FROMLINE, ENDRPE, COPYRANGE
FORMAT and PDE	PORT, LAND, IPORT, ILAND, SETMARGIN, SETFONT, INDEXFONT

LCDS COMMANDS	VIPP® COMMANDS
NUMBER	SETPAGENUMBER
PAPERSIZE	SETPAGESIZE
RESOLUTION	SETRES
SHIFT	SHIFT
STAPLE	SETFINISHING, ENDOFRUN
UNITS	SETUNIT
DATA	FROMLINE, recpos length
FONTINDEX	SETRPEPREFIX, RPEKEY, RPEDEF
MARGIN	FROMLINE, RPEKEY, Xpos
PCC and PCCTYPE	BEGINPCC, ENCPCC, SETPCC, SETSKIP
VFU, BOF, TOF	SETVFU, SETGRID, SETSKIP
ROUTE, RTEXT, RFORM	SLIPSHEET, SETFORM, CPCOUNT, BEGINPAGE
IDEN	PROCESSDJDE
JDE and JDL	SETJDT

## CONVERSION EXAMPLES

This section provides some LCDS code and VIPP® code examples that demonstrate the conversion issues contained in the previous sections.

### OUTPUT and RTEXT

This information illustrates the conversion of OUTPUT and ROUTE LCDS statements to VIPP®.

#### LCDS code

```

OUTPUT  FORMS=FORM23, COPIES=2, COVER=FRONT;
ROUTER  FORM=ROUT23,
        RTEXT=('Message 1',1,10,5,3),
        RTEXT=('Message 2',2,10,5,3);

```

## VIPP® code

```
2 SETCYCLECOPY
(form23.frm) SETFORM
/VAR_CPC 0 SETVAR
{ IF VAR_CPC CPCOUNT ne          % new copy set
  { { 2 SETMAXFORM
    (rout23.frm) SETFORM
    { ORITL 600 300 MOVETO
    /NCR 20 SETFONT
    CASE CPCOUNT {
      1 { (Message 1) SH }
      2 { (Message 2) SH }
    }
    ENDCASE
  } 1 SETFORM
  (AUX) SETMEDIA
} SLIPSHEET
/VAR_CPC CPCOUNT SETVAR
} ENDIF
} BEGINPAGE
```

## CME

This information illustrates the conversion of a CME LCDS statement to a VIPP® RPE.

## LCDS code

```
CME1:CME LINE=(1,3),POS=1, FONT=1,
POS=51, FONT=2,
  LINE=(4,-),POS=1, FONT=3;
CME2:CME LINE=(1,3),POS=1, FONT=1,
POS=51, FONT=2,
  LINE=(4,-),POS=1, FONT=3,
POS=81, CON=(*****);
PDE1:PDE PMODE=PORTRAIT, BEGIN=(2.4 CM, 3 CM)
  FONTS=(HE10BP,HB10BP,HE08BP);
OUTPUTCOPIES=2, FORMAT=PDE1, MODIFY=(CME1,1,1),MODIFY=(CME2,2,1);
```

## VIPP® code

```
CM SETUNIT
PORT
2 SETCYCLECOPY
2.4 0 3 0 SETMARGIN
/F1 /NHE 10 INDEXFONT
/F2 /NHEB 10 INDEXFONT
/F3 /NHE 8 INDEXFONT
2 BEGINRPE
1 FROMLINE
  [ 0 0 0 nu]] 0 .5 0 50 /F1 BLACK ]
  [ 7 0 0 nu]] 0 0 50 82 /F2 BLACK ]
4 FROMLINE [1] COPYRANGE
  [ 0 0 0 nu]] 1.5 .4 0 132 /F3 BLACK ]
4 FROMLINE [2] COPYRANGE
  [ 0 0 0 nu]] 1.5 .4 0 80 /F3 BLACK ]
  [ 7 0 0 nu]] 0 0 0 (***** ) /F3 BLACK ]
ENDRPE
```

## RPAGE

This information illustrates the conversion of RPAGE LCDS statements to VIPP®.

**LCDS code (WHEN=TOP)**

```
T1:TABLE CONSTANT='Page 1';
C1:CRITERIA CONSTANT=(100,7,EQ,T1);
RPAGE TEST=C1, SIDE=NUFRONT, WHEN=TOP;
```

**VIPP® code**

```
/TEST1 1 66 100 7 /eq (Page 1) SETPCD
{ IF TEST1 { NEWFRONT } ENDIF } BEGINPAGE
```

**LCDS code (WHEN=BOTTOM)**

```
RPAGE TEST=C1, SIDE=NUBACK, WHEN=BOTTOM;
```

**VIPP® code**

```
{ IF TEST1 { NEWBACK } ENDIF } /P ENDPAGE
```

**LCDS code (WHEN=NOW)**

```
RPAGE TEST=C1, SIDE=NUFRONT, WHEN=NOW;
```

**VIPP® code (VIPP® 2001 and later)**

```
/TEST2 100 7 /eq (Page 1) SETPCD
/TEST3 1 1 100 7 /eq (Page 1) SETPCD
[ /TEST2 [/SK1 true 1 0 0] ] SETSKIP
{ IF TEST3 { NEWFRONT } ENDIF } BEGINPAGE
```

**Translating DJDE records**

The PROCESSDJDE command simplifies the conversion of DJDEs in a JDT. DJDEBEGIN can be used inside the PROCESSDJDE command to process LCDS BEGIN statements.

```
{ CASE DJDECMD {}% default action = none
  (JDL) { DJDEPAR SETJDT }
  (BEGIN) { DJDEPAR DJDEBEGIN }
  (FORM) { DJDEPAR SETFORM }
  (FEED) { CASE DJDEPAR {
    (AUX) {(Preprinted) SETMEDIA}
    (MAIN) {(Plain) SETMEDIA}
  }
  ENDCASE
}
} 0 ($$DJDE) 3 PROCESSDJDE
```

This is an example of how PROCESSDJDE is used:

**Standard SYSOUT printing**

```
%!
ILAND
132 66 SETGRID
/NCR 0 SETFONT
DUPLEX_on
(form1.frm) SETFORM
```

This is an example of the VIPP® code used to print standard sysout:

**PCC bytes**

VI Compose handles PCC bytes using the VIPP® SETPCC command. Customized PCC tables can be defined using BEGINPCC and ENDPCC. Predefined standard PCC tables are defined in `xfi/src/xfi.pcc (/ANSI,`

/IBM1403, /IBM1403A).

## Font index

LCDS FONTINDEX command can be handled by VIPP® as follows:

- Using SETRCD tests inside FROMLINE/RPE tables
- Using SETRPEPREFIX and RPEKEY/RPE tables
- Using RPEDEF

## LCDS MIGRATION HINTS AND TIPS

This section contains LCDS migration hints and tips.

### Comparable FDL section commands in VIPP®

When using comparable FDL section commands in VIPP®:

- When using FDL section commands to build a Multi-Up form, write the VIPP® form code as you would for one-up page signatures, and use the **SETMULTIUP** command in the JDT. This action sets up scaling and places the form on each page.
- To repeat a section in the form, define a local segment with the **XGFRESDEF** command, and call it with the **SCALL** command.

Example:

```
%!  
%%Title: formxyz.frm  
{  
/SEC1  
{ 50 50 MOVETO (test) SH  
  0 100 100 100  S1 DRAWB  
} XGFRESDEF  
.....  
100 100 MOVETO (SEC1) SCALL  
500 500 MOVETO (SEC1) SCALL  
....  
} FSHOW
```

### Converting LCDS resources to VIPP® resources

To convert legacy resources, it is recommended that you use a third-party tool. If you require assistance with the resource conversion, contact a local Xerox representative.

### Handling host banner pages

To handle host banner pages, use the second syntax of the **STARTLM** command. The following example shows how `banner.jdt` is used to format the first three pages of the data file, and how to use `data.jdt` to form at the remainder of the data file. For example: `(data.jdt) (banner.jdt) 3 STARTLM`

Another method is to use the **SETPCD** and **BEGINPAGE** commands.

### Testing for banner criteria

Banner detection and associated actions can be defined using the **SETPCD**, **IF**, **ELSE**, **ENDIF**, and **BEGINPAGE** commands. These commands allow you to test for banner criteria in the data for your job or for report separation purposes. Include the following code in a JDT:

```
/BANNER 1 5 0 100 /eq 100 (*) SETPCD  
{ IF BANNER { (:red) SETMEDIAT } ENDIF } BEGINPAGE
```



# Specialty Imaging with VIPP®

This chapter contains:

Specialty Imaging Features.....	208
Programming Support.....	210
Using Specialty Imaging .....	211
Demonstration Jobs .....	216
Using Specialty Imaging Effects .....	217
Limitations .....	221
Design Considerations for Correlation Mark.....	223
Design Considerations for FluorescentMark Color .....	227
Design Considerations for GlossMark® Text .....	236
Design Considerations for Infrared Color .....	239
Design Considerations for MicroText Font .....	244
Design Considerations for Void Pantograph .....	247

VI Compose provides support for variable Specialty Imaging. Specialty Imaging is a reference to an area of specialty printing in the offset industry. It covers special printing effects as described below.

Xerox Specialty Imaging effects are only supported on Xerox toner based printers (print engines) using FreeFlow Print Server or EFI controllers (DFE's). These effects are not supported on other printers or other controllers. No support for Xerox Specialty Imaging if offered when using VI Compose (Open Edition). There are five Specialty Imaging effects, plus additional extensions of these effects using *Patterned Ink*. Not all effects are supported on all Xerox print engines and not all colors or fonts will print the same across the different print engines.

Specialty Imaging effects can be used to provide low-cost document security or novelty effects designed to grab the reader's attention. While the effects maybe reproducible on a good copier, by combining these effects with variable data, it becomes much more difficult to change or modify any of the Specialty Imaging effects without destroying the visual appearance. In addition, some effects can be hidden or are not obvious to the casual observer. For example the Fluorescent or Infrared effects.

Specialty Imaging effects are especially useful in applications, such as parking passes, event tickets, ID Cards or other types of documents, that would benefit from some level of fraud protection. Xerox VIPP® Specialty Imaging can provide this added protection at a low cost, without the need to add additional hardware, toner or expensive post processing equipment.

VIPP® Specialty Imaging effects add the ability to enhance the design by including text or in most cases except with MicroText or SI Patterned Ink bi-level images that can only be viewed under special circumstances. These effects are achieved through the use of color, media and Xerox technology. Specialty Imaging effects can only be reproduced when printing the VIPP® application, they are not reproducible when viewing on the screen.

Specialty Imaging effects are based on technologies used to embed an alphanumeric string into a print pattern in such a way that the string is indecipherable under normal viewing conditions, but becomes visible under special viewing conditions, or with the aid of simple tools such as a loupe, UV light, or infrared camera.

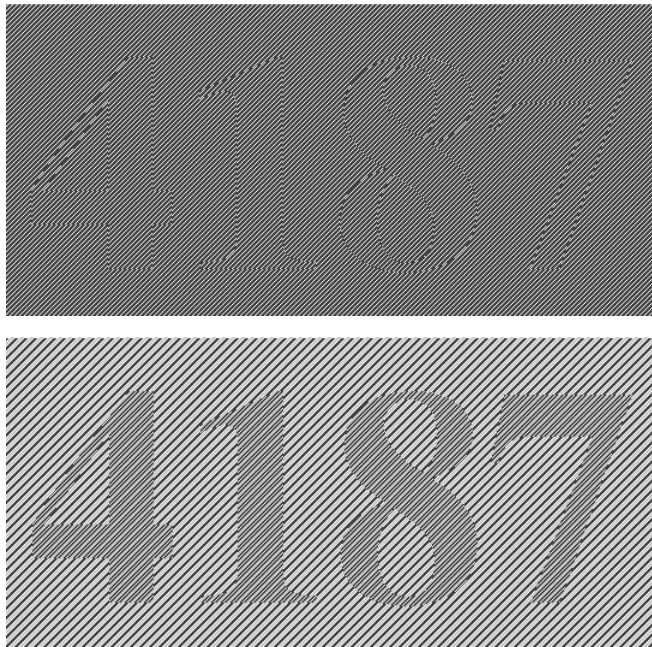
Due to the unique properties of the Specialty imaging fonts and colors, not all fonts and SI colors will work equally well across all Xerox production devices nor on all types of media. It is highly recommended that you use any of the VI design tools to code a SI patch book using the effects you are interested in. Use a combination of different font families and SI colors. Print this patch book to the target print device and review the output. This will help you select the effects and SI colors that work best for your application on the target print device.

VI Compose includes support for printing text or in most cases except with MicroText or SI Patterned Ink bi-level images using the following Specialty Imaging elements:

### Correlation Mark

Correlation Mark is used to print text or bi-level images in a way that the content is not visible unless it is superimposed by a key transparency. A two-layer CorrelationMark effect is also available where two strings can be embedded into the effect. When the key is used one way the first message is viewable, if the key is flipped or rotated the second string becomes visible.

These two figures show an example of a Correlation Mark application. The first image is the view without the key transparency, the second image is the view with a key transparency overlay. CorrelationMark fonts work best with constant colors, but color gradations can be used. The CorrelationMark Key can be created by printing the SI\_VP\_Correlation\_key.nm file on a transparency. This file can be found in the `xgf/demo` folder.



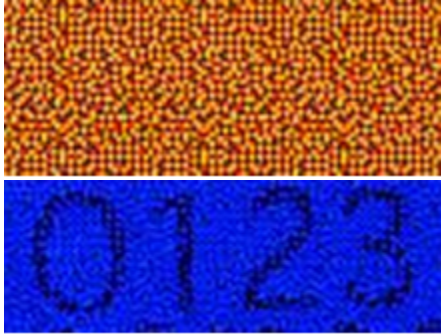
### FluorescentMark color

FluorescentMark color is used to print text or bi-level images in a way that the content is invisible under the normal light but becomes visible under UV / black-light. A two-layer Fluorescent effect is also available where a visible second string can be printed on top of the effect. When a UV light is used to illuminate the effect the visible string fades to reveal the underlying hidden text string.



Note: The media upon which the effect is printed is very important to the effect. Each media must be validated.

These two figures show a color patch under the normal illumination (top) and under UV illumination (bottom). Note that no special material other than UV light, is used to view the hidden text.



### GlossMark®Text

GlossMark®Text is used to print text or bi-level images in a way that the content is not visible in straight-on view, but becomes visible as gloss under inclined illumination, by tilting the printed page.



Note: Some printers specifically have matte toner, which greatly lowers the possible gloss effect.

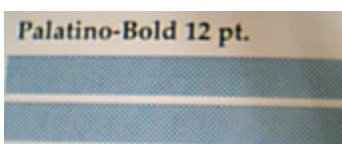
These two figures show GlossMark Text produced on standard mode coated paper. The first image is in straight-on view. GlossMark Text works best with constant colors, but color gradations can be used.

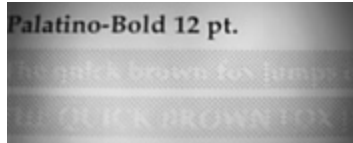


### Infrared color

Infrared color is used to print text or bi-level images in a way that the content is invisible under the normal light but becomes visible under infrared light. A two-layer Infrared effect is also available where a visible second string can be printed on top of the effect. This top string disappears when viewed under IR light conditions, allowing source is used to illuminate the effect the visible string fades to reveal the underlying hidden text string to be viewed.

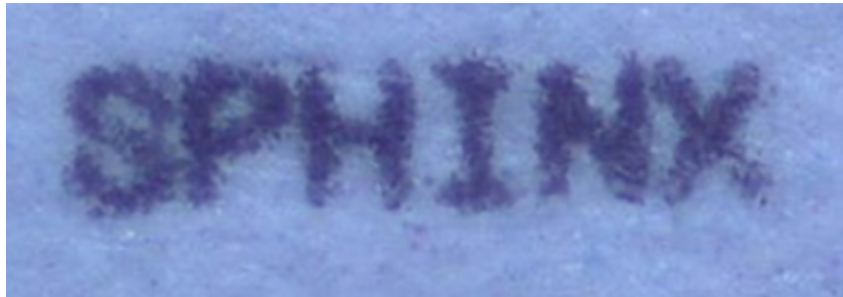
These two figures show a color patch under normal illumination (top) and under Infrared illumination (bottom). The illustration on the bottom was captured using a web cam and displayed on a Windows computer.





### MicroText font

is used to print point sizes normally smaller than one point. The top figure shows the quality of a MicroText font printed on standard coated paper. For comparison, in the bottom figure, the microprinting on Japanese currency is shown at identical scale. When judging quality, it is useful to compare the results to currency printing.

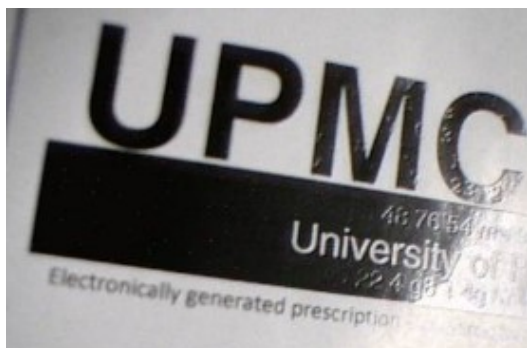


### Artistic MicroText

provides the ability to paint an object with a string of microtext. Microtext being characters smaller than one point.

### Artistic Black

provides the ability to paint an object using a gloss or matt black background with a repeating text strings or a bi-level image. When the background is matte, the string will be gloss, when the background is gloss then the repeating text string is matte. Coated paper is recommended. Artistic Black does not copy.



## SI Patterned Ink

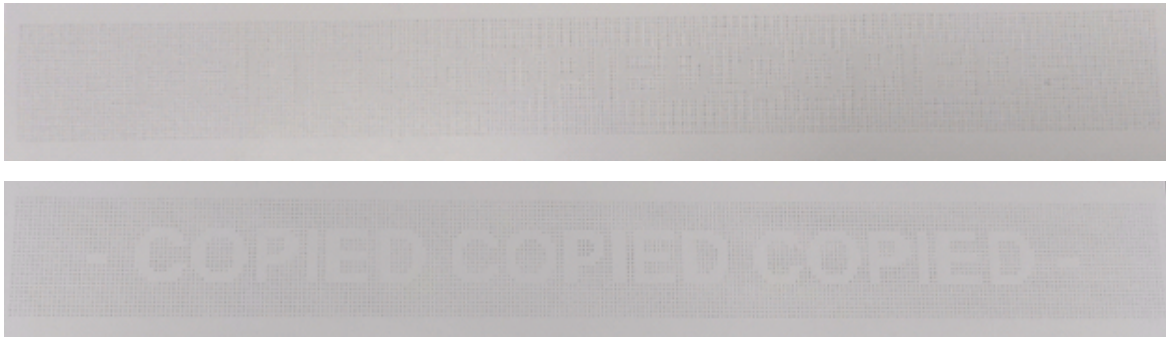
provides the ability to paint a repeating text string with an overlaying SI effect

- Fluorescent
- Gloss
- Infrared
- Correlation

Specialty Imaging elements can be used to embed information in a document that pertains to its identity, authenticity, value, ownership, or any other element that is of value to the rightful owner of the document. The unique printing capabilities of Specialty Imaging can be used to deter fraud and add visual effects to the document's layout. Specialty Imaging is accomplished through special Xerox fonts, VIPP® Colorkeys, and extensions to the VIPP® SHx commands. All SHx commands work except for the SHX command. SHx refers to SH, SHP, SHMF, and related commands. The color/ GEPkeys are delivered as part of VI Compose installation, however, the Xerox Specialty Imaging font sets must be downloaded and see the start of this section for download details to create the special imaging effects. For more information about Specialty Imaging, contact a local Xerox representative.

## Void Pantograph

A void pantograph provides the security and protection to the documents or coupons and so on. This effect to be achieved by printing the text which is filled by pixels on top of the background pixels. The intention is to hide the text in the background pixels in the original document and the text cannot be visible to the human eyes. If the document which is protected with void pantograph effect is duplicated, then hidden text is clearly visible to the human eyes. The effect looks like as in the following images.



Note: Specialty Imaging Printer fonts, the Correlation key and other SI files can be downloaded, Refer to [Specialty Imaging and Barcode Font download](#) for more information.

## Specialty Imaging Features

**Correlation Mark** refers to printing variable text strings or a bi-level images in a way that the content is not visible unless it is superimposed by a "key" transparency. A second string can also be added that is visible when the transparency key is rotated 90 degrees. The "key" printed on a transparency will need to be supplied to all end-users that need to read the data. The Correlation Mark "key" can be created by printing the file, SI\_VP\_Correlation\_key.nm, on a transparency. A Correlation Mark Color Patch book (SI\_CorrColorPatchBook.ps) is available in xgf/demo. Correlation effects can be defined using the SETPAT command (/VPCR option).

**FluorescentMark color** refers to printing variable strings or bi-level images in a way that the content is not visible under normal light but becomes visible under UV / black-light. A special top layer text string can be printed on the Fluorescent effect that is visible under normal light but will fade allowing the underlying fluorescent text to become visible under UV light. VIPP® has built-in pre-defined FluorescentMark (sometimes referred to as UV) Colorkeys; these are listed in a table at the end of this section. To produce the effect use these Colorkeys in pairs, one for the background and one for the font color. The table shows the pairings available.

Any available font can be used as the text in the FluorescentMark color effect, for best results a small point size (such as 14pt. or below) should be used.

The clarity of the FluorescentMark colors depends on many factors, including the paper used for the application. As with all Specialty Imaging applications, test the Specialty Imaging application to ensure it meets design and performance requirements.

**GlossMark Text** refers to printing variable text strings in a way that the content is not visible in straight-on view, but becomes visible as gloss under inclined illumination. VIPP® has built-in pre-defined GlossMark Colorkeys; these are listed in a table at the end of this section. Custom colors can also be defined. A Gloss Color Patch book (SI\_GlossColorPatchBook.ps) is available in xgf/demo. GlossMark effects can be defined using the SETPAT command (/VPGL option).

The clarity of the GlossMark Text depends on many factors including the paper used for the application, and the toner used on the device. GlossMark Text makes use of the glossiness in the toner. Some low gloss EA toners may affect the ability to create GlossMark Text effects. Some printers have settings that can be used to tone up or down the gloss of the toner. As with all Specialty Imaging applications, test the Specialty Imaging application to ensure it meets design and performance requirements.

**Infrared color** refers to printing variable strings in a way that the content is not visible under normal light, but can be viewed when using Infrared light. Using a special IR color, visible text can be printed that is visible under normal light conditions but fades to display the hidden text when viewed under IR lighting conditions. Predefined Infrared Colorkeys are listed in a table at the end of this section.

**MicroText font** refers to printing variable text strings at point sizes normally smaller than one point. Typically you would print a string that repeats the information several times on a line or in a paragraph.

The clarity of the MicroText font depends on many factors including the smoothness of the paper used for the application. As with all Specialty Imaging applications, test the Specialty Imaging application to ensure it meets design and performance requirements.

**Artistic MicroText** is similar to patterned ink described below where a repeating text string is used to fill (paint) an object. However the repeating text effect uses one of the available microtext fonts.

**Artistic Black** uses a gloss or matte background color with a repeating text string. When the background is gloss, the text will be matte. When the background is matte, then the text will be printed using a gloss effect.

**Patterned Ink** is a text based ink effect where a background color (optional) and a foreground repeating text string

is used to fill (paint) an object. Patterned ink can be defined using the **SETTPAT** command. The background color can be a GEPkey selection or can be a combination of SI effect and SI color, such as Gloss or Correlation.

**Void Pantograph** is the special effect based on the text pattern. This effect works with combination of background pattern and foreground text pattern. Background pattern defines the size and frequency of the pixel for the background patch that hides the text. Foreground text pattern defines the size and frequency of the pixel for the text that needs to be hidden. To define this pattern, use the **SETPAT** command. Background pattern and foreground text pattern are set with values in **SETPAT** command.

### Specialty Imaging adds complexity to page processing

The use of Specialty Imaging can add a level of complexity to page processing at the printer that in certain circumstances can exceed the limitations of the printer. Therefore, the effects cannot be guaranteed to work in all applications. The number of specialty imaging effects on a page, the size of the effect and the media can affect the appearance of the printed output. It is recommended that you test the application before you run the job in production mode.

### Specialty Imaging Effects and Fraud Protection

Do not depend solely on Specialty Imaging to protect high value documents against fraud. Specialty Imaging allows you to add low levels of protection to your document by embedding variable data into the effects. The effects cannot be changed without destroying the look and feel of the effect.

### Specialty Imaging Support

Xerox Specialty Imaging is supported on Xerox Production print engines and on some Xerox Office Print engines.

## Programming Support

Programming support for Specialty Imaging features is built in to the software. You have access to a set of built-in variables and graphical element property (GEP) keys.

### MPR

A built-in variable used with MicroText fonts.

### Built-in Colorkeys

Pre-defined colors are available for use with GlossMark, Correlation Mark, Infrared colors, and FluorescentMark colors. For information on GlossMark custom colors, refer to the *Gloss Color Patch* book, *SI\_GlossColorPatchBook.ps*, available in `xgf/demo`.

You can download the Xerox Specialty Imaging printer fonts. For download information, refer to the beginning of this section.

GEP keys in `xgf.gep` enable support for FluorescentMark color, Infrared color, GlossMark, and Correlation Mark Specialty Imaging features.

You can use the built-in GEP keys to access feature colors. Each GEPkey invokes a different pattern as its color, which fills a graphic element (i.e. a box, a circle or ellipse, a polygon, etc.) when one of these drawing commands (DRAWB, DRAWC, DRAWPOL, etc.) are invoked. The GEP keys set the color to the associated pattern automatically. Any subsequent text or bi-level image printed on the graphic element is visible.

For lists of the GEP Colorkeys available in `xgf.gep`, refer to one of these sections:

- [Guidelines for creating effective CorrelationMark applications](#)
- [FluorescentMark color tables](#)
- [GlossMark Colorkey table](#)
- [Infrared Colorkey tables](#)

### Example

This example uses the BCALL construct. Use BCALL to isolate the FluorescentMark color text sequence from the rest of the VIPP® Pro code. The construct prints `Hello World` four times in a rectangular box, using `UV_GOLD1` as the background color. The color `UV_GOLD2` is assigned to the printed text string automatically. The text becomes more visible under fluorescent lighting.

```
%% UV Text
{
/NHE 14 SETFONT
300 1400 2100 100 UV_GOLD1 DRAWB
549 1331 MOVETO
[ 4 ( Hello World ) VSUB ] 0 SHP
} BCALL
```

## Using Specialty Imaging

To use MicroText Xerox Specialty Imaging, you can be required to install the MicroText printer fonts on the target print device. Specialty Imaging printer fonts are only required for MicroText effects. Xerox Specialty Imaging fonts are available at [Xerox.com](http://Xerox.com). SpecialtyImaging Fluorescent and Infrared effects which are based on UV and IR colors, and Correlation and GlossMark effects which are based on patterns defined by the **SETPAT** command can use standard fonts available on your device.

Some effects require a specific printer setup. For FFPS, the settings are discussed in this document. For EFI, contact the print vender for specific information on device configuration

### DOWNLOAD AND INSTALL THE SPECIALTY IMAGING FONTS

Specialty Imaging fonts are provided [Xerox.com/SpecialtyImaging](http://Xerox.com/SpecialtyImaging). For more information, refer to [Specialty Imaging and Barcode Font download](#). However, to download Specialty Imaging fonts, you first review and agree to the End User License Agreement that will be presented. If you do not accept the conditions, you will not be able to download the printer fonts.

MicroText fonts are to use with text applications using MicroText. A typical micro font application may be a financial document where critical information such as the account number, check value, etc., may be printed in a selected area multiple times using the micro font. MicroText fonts only support solid color values, such as black, as the font color. This font set can be installed on any monochrome or color system.

Once you have obtained the Specialty Imaging fonts, use the option to Install PostScript Printer fonts provided by your printer. You can refer to the printer documentation.

### SET THE FREEFLOW PRINT SERVER TO USE SPECIALTY IMAGING

The set up for the FreeFlow Print Server controller is of paramount importance when using Specialty Imaging. When you do not use the settings shown below, the application may not print correctly.

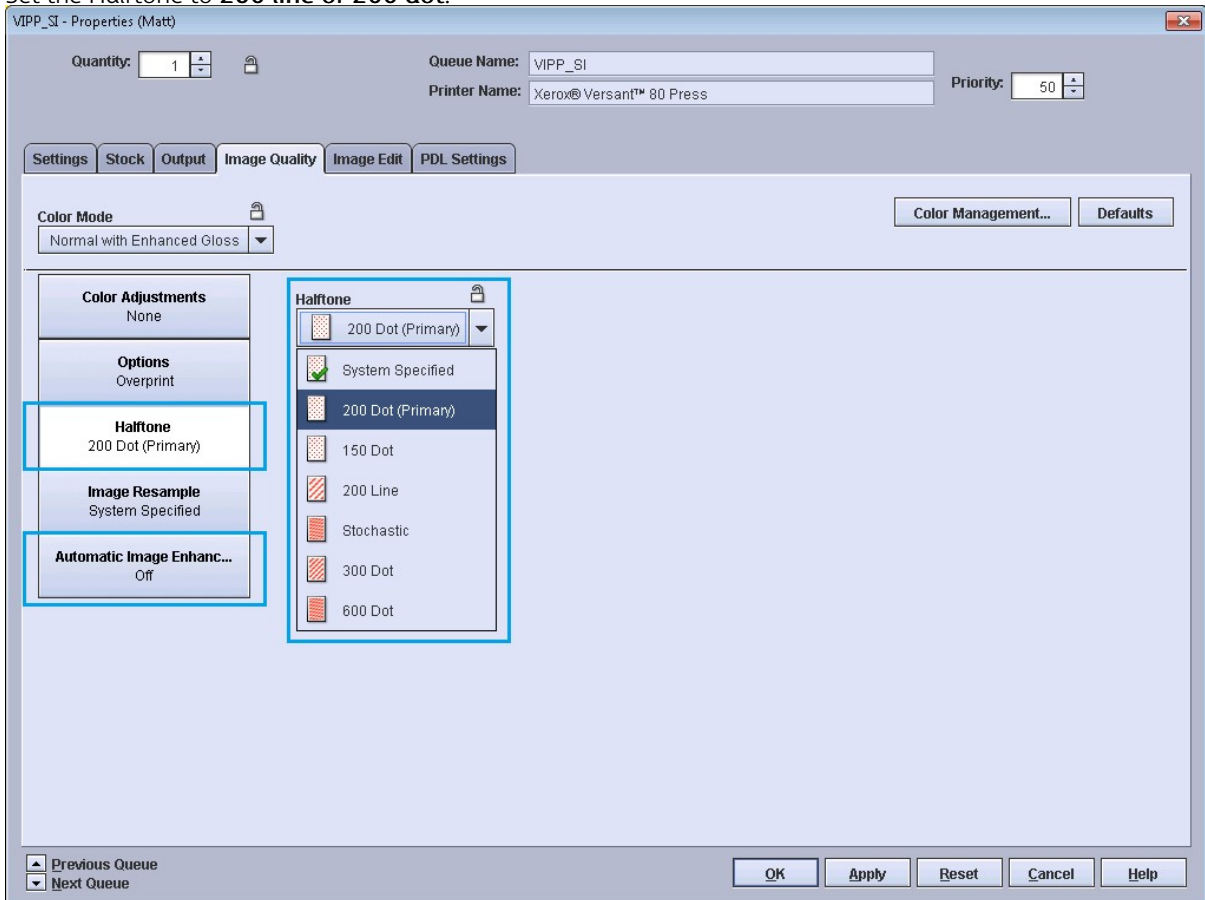
The sections below provide information about color management settings on the FreeFlow Print Server. For EFI settings on the EFI controllers, refer to the documentation supplied with the device, or contact the vendor support site.

#### FreeFlow Print Server 7.0 (and above) Settings

All Specialty Imaging effects, such as, MicroText, Correlation Mark, GlossMark Text, ArtisticBlack Text, Infrared, and Fluorescent require color management settings on the controller. When using one of these colors it is critical that you follow the steps outlined below to set up color management.

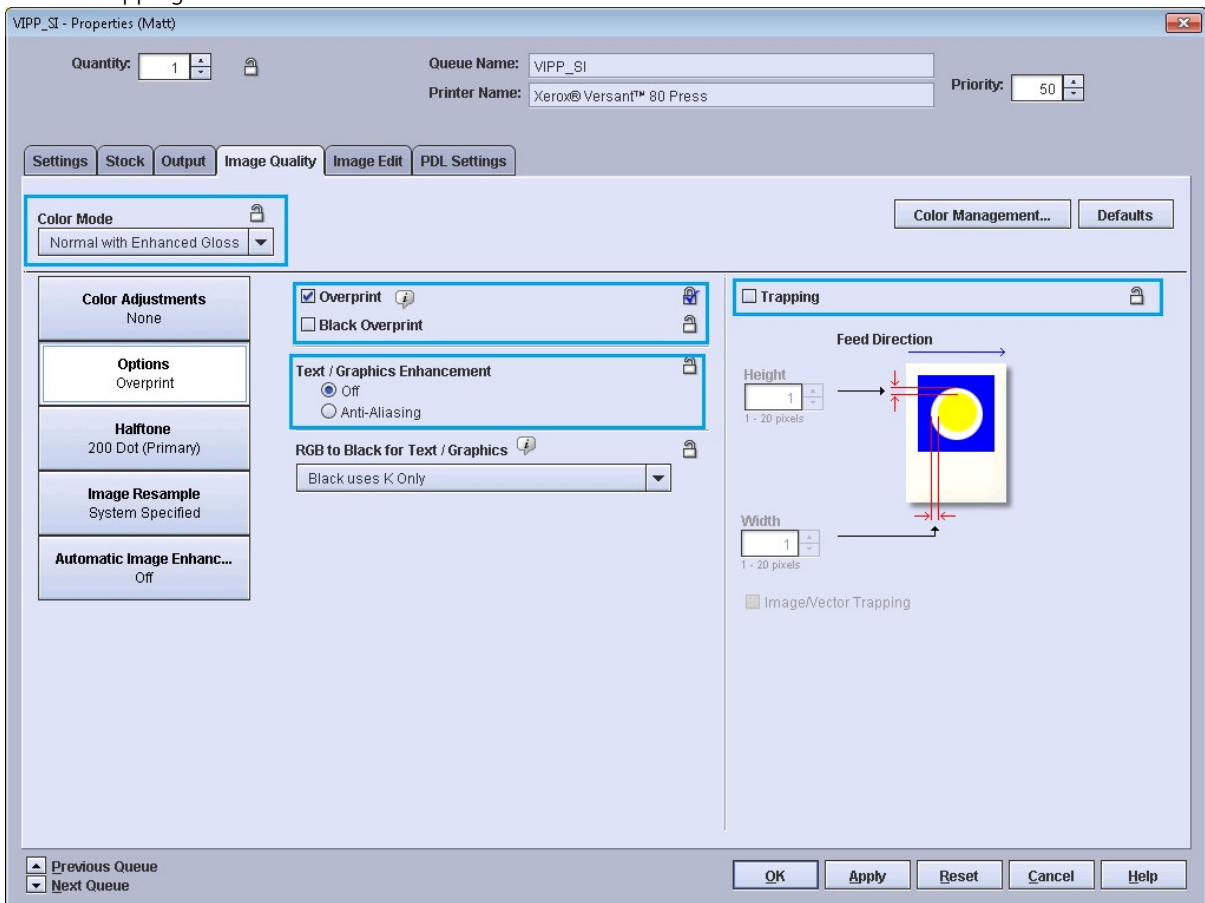
1. Edit the queue settings and select the **Image Quality** tab.
2. Set the Automatic Image Enhancement to **Off**.

- Set the Halftone to **200 line or 200 dot**.



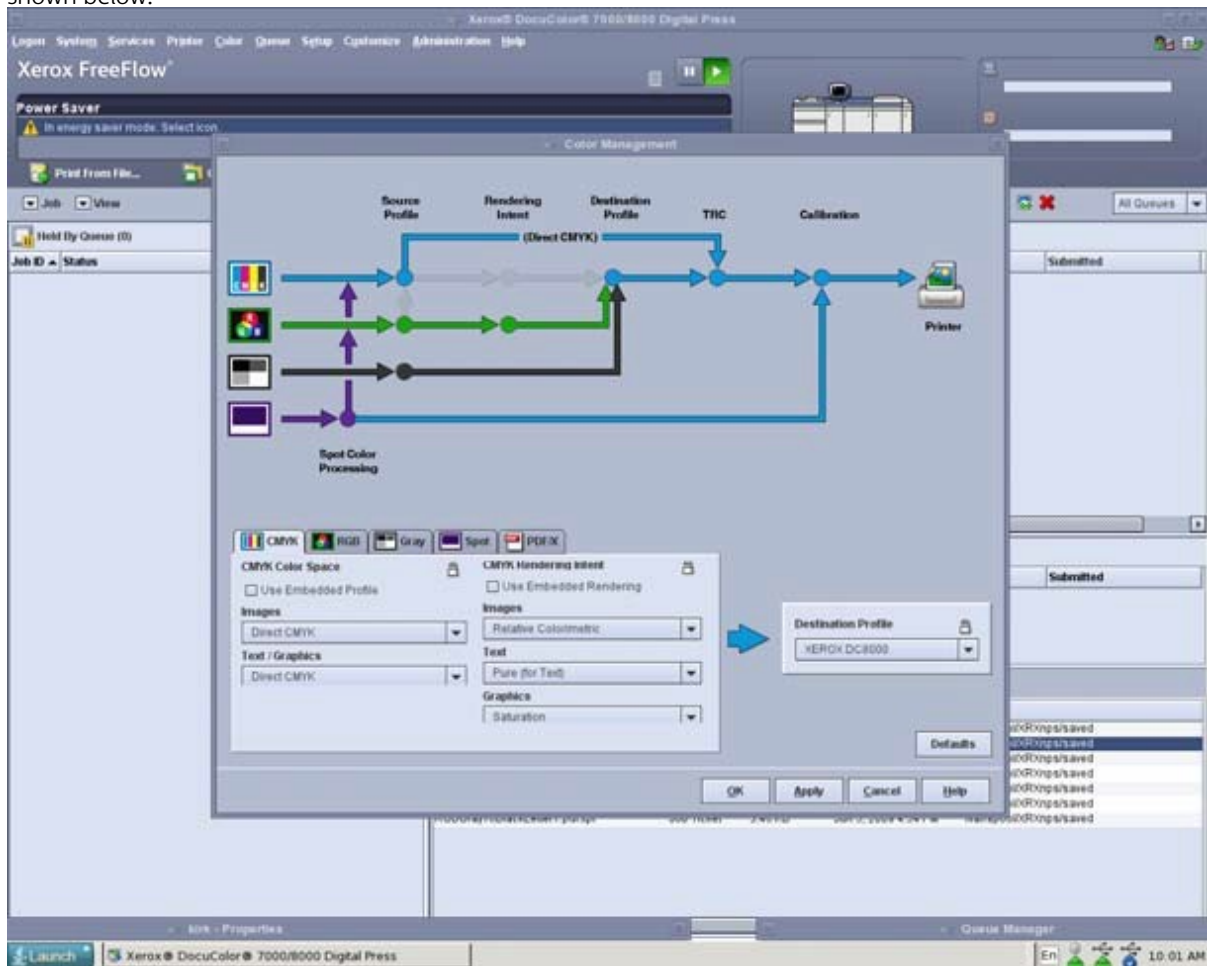
- Set the Color Mode to **Normal with Enhanced Gloss**. (Only for **ArtisticBlack** and **GlossMark** effect)
- Select the **Options** tab in the same menu, check the **PostScript Overprint = On (Override Lock = On)** and **Black Overprint = Off** Option as shown below.
- Set the Anti-Aliasing to **Off**. (With the exception of **On** for MicroText effect)

7. Set the Trapping to **Off**.



8. Select the **Color Management** option to set the CMYK Rendering Intent setting.

9. When using Specialty Imaging Swatch colors use “**Direct CMYK**” for color management, set this option as shown below.



## FFPS PRINTER SUPPORT RESTRICTIONS

Specialty Imaging effects are subject to the following limitations:

- GlossMark Text is not recommended with the DocuColor 242/252/260 or EA Toner based products.
- MicroText Mark is not recommended with the DocuTech 180 HLC, DocuTech 180 HLC Publisher, DocuTech 61XX or DocuPrint family of printers.
- CorrelationMark is not applicable to the Xerox 4110/4590/4595, DocuTech180 HLC, DocuTech180 HLC Publisher, DocuTech 61XX or DocuPrint family of printers
- Venezia Mode is recommended when using FluorescentMark, Infrared Text, or GlossMark Text with the DocuColor 7000AP/8000AP/7002/8002/8080.
- GlossMark Text, FluorescentMark, and Infrared Text are not applicable to the Xerox Nuvera 100/ 120/144 EA DPS, Xerox Nuvera 288 Digital Perfecting System, 4110/4590/4595, DocuTech180 HLC, DocuTech180 HLC Publisher, DocuTech 61XX or DocuPrint family of printers.
- GlossMark Text, FluorescentMark, Infrared Text, Correlation Mark, and MicroText Mark are not applicable to the Continuous Feed family of printers.

- ArtisticBlack Text is not recommended with the Xerox iGen 5 Press with Matte Dry Ink and iGen 150 Press with Matte Dry Ink.
- ArtisticBlack Text, GlossMark Text, FluorescentMark, and Infrared Text are not applicable to the Xerox Brenva HD Production Inkjet Press, Xerox iPrint Production Inkjet System, Xerox Nuvera 288 EA Perfecting Production System, Xerox D136 Copier/Printer.
- GlossMark Text is recommended to be used in 220 or above gsm media.
- Image Quality->Color Mode->Normal with Enhanced Gloss is recommended when using GlossMark Text with the Xerox Color 70 Printer family and Xerox Versant 80 Press.
- Anti-aliasing is recommended when using MicroText with the Xerox Color 1000 Press, Xerox Color 70 Printer, Xerox iGen 5 Press, Xerox iGen 150 Press, Xerox Nuvera 288 EA Perfecting Production System, Xerox D136 Copier/Printer Xerox Versant 80 Press, Xerox Versant 180 Press, Xerox Versant 3100 Press, Xerox Color C75 Press, Xerox Brenva HD Production Inkjet Press, Xerox iPrint Production Inkjet System.

## Demonstration Jobs

The following demonstration jobs (Patch book files) can be found in the /usr/xgfi/demo or D:\xgfi\demo folder. It is recommended that you print these files as they will provide some insight to supported colors or effects on your print device.

<b>SI_CR_2L_Samples.nm</b>	Produces samples of text and colors used in two layer CorrelationMark applications.
<b>SI_IR_2L_Samples.nm</b>	Produces samples of text and colors used in two-layer Infrared color applications.
<b>SI_I R_Samples.nm</b>	Produces samples of text and colors used in single layer Infrared color applications.
<b>SI_UV_Samples.nm</b>	Produces samples of text and colors used in FluorescentMark color applications.
<b>SI_text_patterns.vpc</b>	Produces samples of Artistic MicroText, Artistic Black and patterned Ink samples.
<b>SI_CorrColorPatchBook.ps</b>	Produces a color patch book for Correlation effects.
<b>SI_GlossColorPatchBook.ps</b>	Produces samples of text and colors used in GlossMark color applications.
<b>SI_VP_GlossMark.vpc</b>	Produces samples of text and colors used in GlossMark color applications.
<b>SI_VP_Correlation_1L.vpc</b>	Produces samples of text and colors used in single layer CorrelationMark applications.
<b>SI_VP_Correlation_2L.vpc</b>	Produces samples of text and colors used in two-layer CorrelationMark applications.
<b>SI_VP_Correlation_key.nm</b>	Produces a Correlation Mark key used for Correlation effects.
<b>SI_VP_VoidPanto.vpc</b>	Produces a sample void pantograph text pattern on the background.

## Using Specialty Imaging Effects

Specialty Imaging is supported through extensions to the VIPP® print commands, a set of specialty Colorkeys and built-in VIPP® variables. Specialty Imaging is enabled for any string by setting the correct specialty font, specialty Colorkey when applicable and calling the appropriate **SHx** command. Refer to the **SHx** commands for more information.

Because the Correlation Specialty Imaging effect is best printed over a white background, it is recommended to use the OWHITE color as the background color in the Correlation effect definition (SETPAT). Refer to [Design considerations for Correlation Mark](#) for more information.

In the code examples below, the specialty imaging code is enclosed within a BCALL procedure, which protects the rest of the VIPP® code from any color changes, etc., defined within the BCALL procedure.

### Correlation Mark Example

```
/CR1 [ /VPCR 1 OWHITE CR_Magenta100 ] 0 0 0 1 1 SETPAT % Define Correlation effect
100 2200 MOVETO

(Correlation Mark example:) SHL

/NHE 14 SETFONT % Select font and size for text
100 2096 2199 100 CR1 DRAWB % Draw correlation box
282 2031 MOVETO

(Hello World) SHc % Print text over box
100 1096 2199 600 CR1 DRAWB
282 1031 MOVETO

(truk.tif) 1.5 0 ICALL
```

### 2-Layer Correlation Mark Example

```
/CR2 [ /
VPCR 1 OWHITE CR_Magenta100 CR_Cyan100 ] 0 0 0 1 1 SETPAT % Define 2L Correlation
effect
100 2200 MOVETO

(2-Layer Correlation Mark example:) SHL
{ /NHEB 18 SETFONT % Set text font size
100 2100 500 90 CR2 DRAWB % Draw correlation box
350 2026 MOVETO (Hello World) SHc % Print text over box - 1st layer
350 MOVEH (Welcome) SHc % Print text over box - 2nd layer
} BCALL
```

### FluorescentMark Color Example

```
100 2200 MOVETO
```

```
(FluorescentMark color example:) SHL
/NHE 14 SETFONT
100 2096 2199 100 UV_GOLD1 DRAWB
282 2031 MOVETO

[ 4 (Hello World) ] 0 SHP
100 1096 2199 600 UV_GOLD1 DRAWB
282 1031 MOVETO

(truk.tif) 1.5 0 ICALL
```

## 2-Layer FluorescentMark Color Example

```
100 2200 MOVETO

(2-Layer FluorescentMark color example:) SHL
{ /NHE 14 SETFONT
100 2096 2100 100 UV_YELLOW1 DRAWB
282 2031 MOVETO

[ 4 ( Hello World ) VSUB ] 0 SHP
[YELLOW UV2L] SETTXC % select color and mask for 2-Layer text
282 2031 MOVETO

[ 4 ( GOOD DAY ) VSUB ] 0 SHP % Print 2 layer text
} BCALL
```

## GlossMark Color Example

```
/GL1 [ /VPGL 1 OWHITE GL_Magenta ] 0 0 0 1 1 SETPAT % Define GlossMark effect
100 2200 MOVETO

(GlossMark color example:) SHL
/NHE 14 SETFONT % Select font and size for text
100 2096 2199 100 GL1 DRAWB % Draw GlossMark box
282 2031 MOVETO

(Hello World) SHc % Print text over box
100 1096 2199 600 GL1 DRAWB
282 1031 MOVETO

(truk.tif) 1.5 0 ICALL
```

## Single Layer Infrared Color Example

```
100 2200 MOVETO
```

```
(Single Layer Infrared Color Example:) SHL
/NHE 14 SETFONT % Select font and size for text
100 2096 2199 100 IR_2L_LIGHTGRAY1 DRAWB % Draw IR box
282 2031 MOVETO % Set text position
(Hello World) SHC % This will be the hidden text layer
100 1096 2199 600 IR_2L_LIGHTGRAY1 DRAWB % Draw IR box
282 1031 MOVETO % Set text position
(truk.tif) 1.5 0 ICALL
```

## 2-Layer Infrared Color Example

```
100 800 MOVETO
(2-Layer Infrared Color Example) SHL
{ /NHEB 20 SETFONT % Start BCALL procedure
88 619 1046 -100 IR_OLIVE1 DRAWB % Set IR area/color
580 654 MOVETO % Set text position
(Welcome to my World) SHC % This will be the hidden text layer
IR_Red SETTXC % Sets top layer IR Text color
596 645 MOVETO % Set text position
(Hello World) SHC % End BCALL Procedure
} BCALL
```

## MicroText Font Example

```
100 400 MOVETO
(MicroText font example:) SHL
[/micro_f9 (micro_F9.fnt)] null SETENCODING
108 321 MOVETO % Set Current Print Position
{ /micro_f9 MPR SETFONT % Call micro font, use built-in
% variable MPR to set size
[10 (Hello World)] 600 0 SHT % Print text string
} BCALL % Protects settings within BCALL
```

## Patterned Ink Code Examples

### Plain (No Specialty Imaging) Patterned Text

```
/PlainPat1 ($$FNAME. $$LNAME.) VSUB /NHEB 20 10 8 30 [ (RED~.6) (YELLOW~.3) ] SETTPAT
41 606 520 100 PlainPat1 15 DRAWBR
```

### Artistic Black Patterned Text

```
/ArtisticAPat ($$FNAME. $$LNAME.) VSUB /NHEB 20 5 5 45 ARTBLACK_A SETTPAT
/ArtisticBPat ($$FNAME. $$LNAME.) VSUB /NHEB 20 5 5 45 ARTBLACK_B SETTPAT
42 480 250 90 ArtisticAPat 15 DRAWBR % Square with rounded corners
312 480 250 90 ArtisticBPat DRAWC % Oval
```

### Fluorescent Patterned Text

```
/FluorescentPat ($$FNAME. $$LNAME.) VSUB /NHEB 30 5 2 45 UV_GOLD1 SETTPAT
42 367 250 160 FluorescentPat DRAWC % Oval
```

### MicroText Patterned Text

```
/MicroTextPat ($$FNAME. $$LNAME.) VSUB /micro_f9_b MPR 1 1 45 RED SETTPAT
/NGMB 110 140 SETFONT
305 242 MOVETO (VIPP) 0 MicroTextPat 0 SHX % Filled text using Microtext
```

### GlossMark Text Patterned Text

```
/GL1 [ /VPGL 1 OWHITE GL_Maroon ] 0 0 0 1 1 SETPAT
/GlossPat ($$FNAME. $$LNAME.) VSUB /NTMB 24 0 0 5 GL1 SETTPAT
42 184 250 160 GlossPat DRAWC % Oval
```

### Correlation Patterned Text

```
/CR1 [ /VPCR 1 OWHITE CR_Maroon50 ] 0 0 0 1 1 SETPAT
/CorrPat ($$FNAME. $$LNAME.) VSUB /NTMB 24 0 0 5 CR SETTPAT
42 184 250 160 CorrPat DRAWC % Oval
```

### Void Pantograph Patterned Text

```
438 1859 MOVETO
/VP04 [ /VPPG1 12 39 OWHITE BLACK ] 0 0 0 1 1 SETPAT
384 1354 1327 300 /VP04 DRAWB
/NHEB 50 SETFONT
1032 1130 MOVETO
(VOID MARK) 1221 22 SHP
```

## Limitations

Specialty Imaging offers alternatives to existing offset capabilities. In this context it is important to understand that the detailed effect is different from ordinary offset printing.

Specialty Imaging jobs contain complex data, thus, the DFE will behave as with other complex documents. For example:

- A page in a MicroText font can represent 100 pages of 12pt text. Obviously, the RIP time for the page covered in a MicroText font will be different than the RIP time of a page covered in 12pt text.
- Correlation and GlossMark Text is, in essence, text represented by pattern. A page covered with patterns may have a different RIP time from a page covered in 12pt text.
- High data amounts can affect DFE compression issues. Some DFEs require a minimum compression ratio for communication with the Print Engine. Specialty Imaging in some cases can push those limits. When printer limitations are approached, the effects will be visible on the printed page.
- When printer limits are approached, a higher likelihood exists for color drifts, and so on.

There are certain limitations to all Specialty Imaging features. The following list describes the most common limitations.

- Only the Xerox Specialty Imaging fonts available for download are supported (Refer to the start of this section for download details)
- The Specialty Imaging printer fonts are packaged in several zip files for download. The user must review and agree the End User license Agreement to access the fonts for download.
- Xerox recommends using the available GEPkey and Colorkey values when coding VIPP® applications. If using VI Design Express (Adobe InDesign plug-in) tool, Xerox recommends that you use the available Specialty Imaging swatches that are available. See the Specialty Imaging section in the *FreeFlow VI Design Express Users Guide* for more details.
- MicroText fonts contain only these characters (lower case characters are mapped to upper case characters for printing purposes):

**A through Z, 0 through 9, ! @ # \$ % & \* ( ) - \_ = + [ ] ; : ' " , . ? / € , and ¥.**

- You can use Cyrillic characters used with the normal weight Microtext fonts but not in bold.
- Specialty Imaging can affect printer performance. Specialty Imaging requires more processing power than standard text printing; because of this, Xerox recommends testing of the application and adjusting the use of Specialty Imaging based on the device and performance needs.
- Media Selection will affect the Specialty Imaging effects. FluorescentMark works best on a high brilliance media, while GlossMark and ArtisticBlack text works best on a coated or high gloss media above 220 gsm. Xerox paper, such as the Digital Color Elite family of products, can be used for all Specialty Imaging effects. MicroText, Infrared, and Correlation Mark work on all media.
- Set the printer tray and queue to the correct weight of the media used. Failure to do so can affect the quality of the Specialty Imaging effects.
- The condition of the printer and the color setting can affect the appearance and quality of the effects. It is critical to calibrate the printer and to ensure that the queue settings are defined as defined in [Set the FreeFlow Print Server to use Specialty Imaging](#). Failure to do so will result in poor or no Specialty Imaging effects.

- Not all Specialty Imaging effects will print in a satisfactory state on all devices. Due to the unique characteristics of these effects and the individual device settings, configurations, and media selections, you must test and possibly use alternative supported colors, media, fonts, and font sizes to achieve the desired effect.
- Recommended printer resolution is 600 DPI.
- 1st generation EA Toners do not support GlossMark effects.
- Use of UV Coater has not been validated.
- Use of Xerox Specialty Inks with Specialty Imaging effects has not been validated. Xerox Specialty Inks (XSI) is an option on Xerox DC1000/800 devices supported by FFPS and EFI DFE's.

The paper used for Specialty Imaging needs to have sufficient quality and brightness to support the rendering performed. The Xerox Digital Color Elite Gloss families of paper products meet these requirements.

**FluorescentMark colors**

the paper should have sufficient brilliance, as indicated on the paper wrapper.

**GlossMark Text**

the paper should be coated to have a smooth - preferably glossy - surface.

**MicroText font**

the paper should be sufficiently smooth to allow high resolution rendering.



Note: Limitations specific to individual specialty imaging effects are included in sections that describe the effect. Refer to the relevant section for this important information.

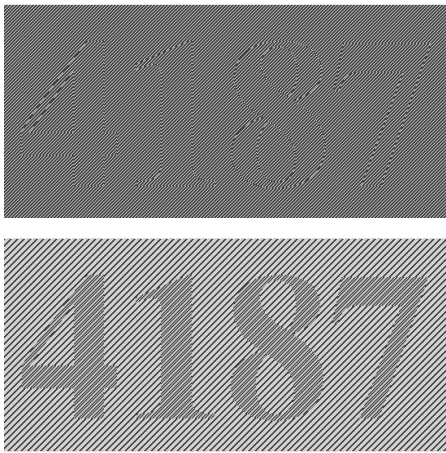
## Design Considerations for Correlation Mark

You find the following information here:

- [Correlation Mark text limitations and requirements](#)
- [Guidelines for creating effective CorrelationMark applications](#)
- [Guidelines for creating effective CorrelationMark applications](#)

The characteristics of Correlation Mark make it useful to use for adding security features to a document. A solid color patch can be used to insert two-layers of extra information.

These two figures show an example of a Correlation Mark Text application. The first image is the view without the key transparency, the second image is the view with a key transparency overlay. Correlation Mark works best with constant colors. The Correlation Mark key can be created by printing the file, SI\_VP\_Correlation\_key.nm on a transparency. This file can be found in the `xgf/demo` folder.



The Correlation Mark feature refers to printing variable text strings in a way that the content is not visible unless it is superimposed by a key transparency. The key printed on a transparency will need to be supplied to all end-users that need to read the data.

Once defined with the **SETPAT** command Correlation Mark effect can be printed using any font available on your printer.

### Note on legacy Correlation fonts

The legacy method of printing Correlation Mark text using a collection of dedicated fonts is deprecated. It is still available and supported for backward compatibility but for new applications it is recommended to use the new Correlation Mark feature which offers more flexibility.

Correlation Mark is available as a single layer or two-layer effect. The single layer prints a single string and will become visible when the Correlation Mark key transparency is placed over the effect. A single layer Correlation effect is defined using a SETPAT statement with the following syntax, refer to the *VIPP® Language Reference Manual* for syntax details:

```
/PATkey [ /VPCR Iwidth coll col2 ] Htr Vtr rot Hscale Yscale SETPAT
```

The two-layer effect uses two text strings, each printed on top of each other. Each layer may have a different color.

A two-layer Correlation effect is defined using a SETPAT statement with the following syntax:

```
/PATkey [ /VPCR lwidth col1 col2 col3 ] Htr Vtr rot Hscale Yscale SETPAT
```

In a two-layer application, one string is visible when the transparency key is used one way. When the key is flipped over or rotated 90 degrees, the second string becomes visible.

The clarity of the Correlation Mark font depends on many factors used for the application. As an example, Xerox Digital Color Elite Cover or Paper stocks provide good results. It is important to set the paper weight correctly at the device to optimize the effects. In particular, media used for GlossMark effect should be above 220 gsm weight. As with all Specialty Imaging applications, test the application to ensure it meets the design and performance requirements.

You can print a Correlation Mark effect anywhere on the document. Because the Correlation effect is best printed over a white background, it is recommended to use the OWHITE color as the background color in the Correlation effect definition col1 argument of SETPAT.

## CORRELATION MARK TEXT LIMITATIONS AND REQUIREMENTS

<b>Paper Requirements</b>	The Xerox Digital Color Elite Gloss families of paper products are recommended.
<b>Text Color</b>	A VIPP® CR_xxx Colorkey is recommended. (Where xxx is InDesign color swatch beginning with CR_ and ending with a color name, for example, GL_Magenta50.)
<b>Text Content</b>	Any arbitrary content. Best results are obtained when using a limited number of characters.
<b>Color</b>	The patch color is defined as DEVICE CMYK, no emulation (SWOP, Fogra, EuroScale, and so on.) can be performed.
<b>DFEs</b>	Xerox print engines driven by FFPS and EFI DFEs. Due to the large amount of data contained in a Correlation Mark, using several Correlation Mark on a single page can cause compression issues.
<b>Color</b>	Select one of the predefined Swatch colors listed in the InDesign Swatch panel. Available Correlation Mark colors are prefixed with CR_. Do not change the Swatch name or the color values as these have no effect on the color used at print time, VI Compose uses pre-defined Colorkeys. Making any changes will invalidate that color selection and unpredictable results may occur.

## CORRELATION MARK COLOR SWATCHES

Single and two-layer Correlation Mark effects can use any of the available predefined color swatches that have the prefix CR\_ in the color swatch panel. The available colors are listed.

COLOR SWATCHES			
CR_Black50	CR_DarkBlue100	CR_Green75	CR_Olive100
CR_Black75	CR_DarkGreen50	CR_Green100	CR_Red50
CR_Black100	CR_DarkGreen75	CR_Magenta50	CR_Red75

COLOR SWATCHES			
CR_Blue50	CR_DarkGreen100	CR_Magenta75	CR_Red100
CR_Blue75	CR_DarkRed50	CR_Magenta100	CR_Teal50
CR_Blue100	CR_DarkRed75	CR_Maroon50	CR_Teal75
CR_Cyan50	CR_DarkRed100	CR_Maroon75	CR_Teal100
CR_Cyan75	CR_Gray50	CR_Maroon100	CR_Yellow50
CR_Cyan100	CR_Gray75	CR_Olive50	(deprecated)
CR_DarkBlue50	CR_Gray100	CR_Olive75	CR_Yellow75 (deprecated)
CR_DarkBlue75	CR_Green50		CR_Yellow100 (deprecated)



Note: Do not change the name or the color settings on the color definitions. The color names listed give an approximation of the actual color pattern that prints on the device.

## Guidelines for Creating Effective CorrelationMark Applications

### Choice of printer

CorrelationMark works best with color printers of EPC and Production Color class.

### Choice of media

CorrelationMark works on all paper types.

### DFE settings

CorrelationMark has been tested with FFPS. Set the image path or queue for printing raw device CMYK.



Note: Do not set press emulation or any other mode that transforms the CMYK colors in the document. Ensure that only the color calibration function is in the image path.

### Choice of CorrelationMark colors

For a wide range of color selections, refer to the *CorrelationMark Patch Book*, SI\_CorrColorPatchBook.ps, available in xgf/demo. Not all choices produce the desired CorrelationMark effect. To determine what colors work best, print the Patch Book on the target printer. For a two-layer color application, refer to the separate patch book that contains a limited set of recommended color choices for two-layer applications.

### Key Transparency

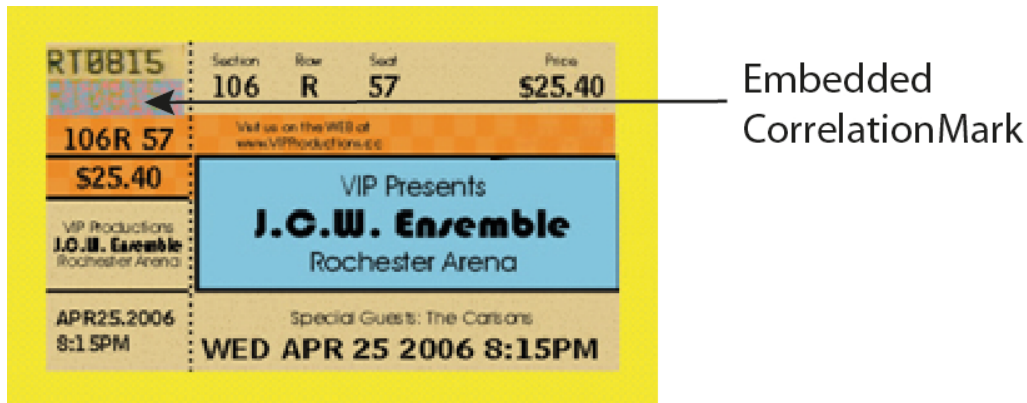
The Two-Layer Key file is a VIPP® file, SI\_VP\_Correlation\_key.nm available in xgf/demo. Print the Two-Layer Key file on a transparency. In the correct orientation, the printed transparency reveals the CorrelationMark Text in each layer.

### Custom Correlation effects

If you vary the SETPAT parameters Htrans, Vtrans, rotate, Hscale, and Yscale, you can create custom correlation effects that work only with the associated key transparency, using the same parameters. You can make copies of the SI\_VP\_Correlation\_key.nm file, then use a text editor or VI Design Pro to create the appropriate key transparencies to match custom effects.

You can use Correlation Mark to embed document information about identity, authenticity, value, ownership, or any element that is of value to the document owner.

In the concert ticket example, the constant color area holds variable data.



You can place other information in the same spot using the 2-layer effect. The top example shows the string ONE, placed in one layer. The bottom example shows the string TWO in the same spot, using a 2-layer effect, and overprinting both texts on top of each other.



## Design Considerations for FluorescentMark Color

FluorescentMark color, available on FFPS and EFI full color Xerox engines, refers to printing variable text in a way that the content is not clearly visible under normal light but becomes visible under UV/ black-light.

The two-layer fluorescent effect allows you to further disguise the effect by laying down a layer of text that is visible under normal light conditions, but will fade to reveal the hidden text under UV illumination.

For better Fluorescent effects use a brilliant white media.

As with all Specialty Imaging applications, test the application to validate the effect meets the design specifications and print performance.

### FLUORESCENTMARK LIMITATIONS AND REQUIREMENTS

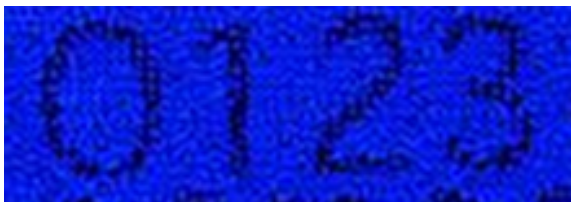
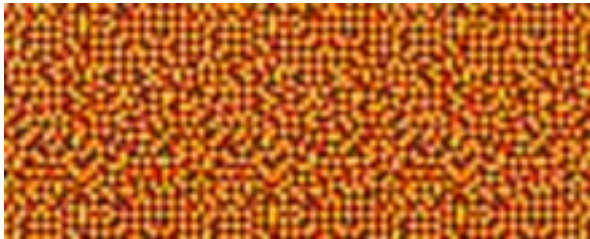
You find the following information here:

- [FluorescentMark color limitations and requirements](#)
- [FlourescentMark color tables](#)

The method by which the FluorescentMark colors are created and rendered makes it difficult to alter electronically or physically, and when designed well, makes it difficult for a user to even realize that information is embedded in textured color patterns. Another benefit of FluorescentMark colors is that to decode the hidden information, only a UV light source is required. Many inexpensive types of UV sources exist (e.g. portable money detectors, flashlights, and even key chains). UV light sources are also commonly used for currency validation at many locations.

A layer of text, visible under normal lighting conditions, can be placed on top of the effect to aid in disguising the true purpose of the effect. This text will fade when viewed under UV illumination, allowing the underlying text to show through. The top layer text must use one of the top layer colors listed in this document.

These two figures show a color patch under normal illumination (top) and under UV illumination (bottom). Note that no special material other than UV light, is used to view the hidden text.



#### FluorescentMark color limitations and requirements

No special materials are involved in this effect. Instead the Optical Brightening and Whitening Agent of the paper substrate is used. Consequently some limitations exist based on the paper choice.

Using FluorescentMark color means that text strings that were invisible or hard to see in normal light appear at a

much higher contrast under Ultra Violet light. Since no special materials are used, the limitation is that the UV component of normal light will already excite some of the fluorescence effect. Also, the characteristics of the physical toner (gloss, etc.) might make it possible to decipher the hidden string in visible light. In order to mask this decoding, the color patch is created with a rough texture.

<b>Paper requirements</b>	The effect strength increases with paper brightness. Paper brightness is normally indicated on the packages. The Xerox Digital Color Elite Gloss families of paper products are recommended.
<b>Patch Color</b>	The color name, when sent to the print device, is matched with a pre-defined Colorkey in VI Compose. Making any changes will invalidate that color selection and unpredictable results may occur.
<b>Text Color</b>	Use the UV Colorkeys supplied by VIPP® and shown in the table in <a href="#">FlourescentMark color tables</a> .
<b>Text Content</b>	Any arbitrary content. Best results are obtained when using a limited number of characters.
<b>Font</b>	Any supported font on the printer or supported by VIPP®.
<b>Two-layer Text Color</b>	Refer to the Color swatches for two-layer Fluorescent effect table below. The text for the visible text layer (Layer 2 - top layer) must be one of the colors listed based on the bottom layer UV_color selected.
<b>Text Frame Fill</b>	The text frame must be filled with an UV_colurname selection from the Swatches panel.
<b>Font</b>	Any supported font on the printer.
<b>Text Size</b>	Not restricted, but should be large enough to be easily visible (10 - 14pts, as larger points point sizes may be visible without UV lighting).
<b>DFEs</b>	Xerox only, FFPS and EFI.
<b>Color</b>	The patch color is defined as DEVICE CMYK, no emulation (SWOP, Fogra, EuroScale, etc.) can be performed. Only 1-dimensional TRCs on the individual CMYK components are allowed. The color name, when sent to the print device, is matched with a pre-defined Colorkey in VI Compose. Making any changes will invalidate that color selection and unpredictable results may occur.
<b>Scale</b>	Scaling will affect the masking, but is allowed
<b>Rotation</b>	Rotation will affect the masking, but is allowed.

#### FlourescentMark color tables

FlourescentMark colorkey pairs are listed in the following table.

FLUORESCENT COLOR NAMES (NO COLOR MANAGEMENT SETTING REQUIRED.)			
UV_APRICOT	UV_FORESTGREEN1	UV_LIGHTLIVE1	UV_REDDISHYELLOW1
UV_MELON	UV_FORESTGREEN2	UV_LIGHTLIVE2	UV_REDDISHYELLOW2

FLUORESCENT COLOR NAMES (NO COLOR MANAGEMENT SETTING REQUIRED.)			
UV_BISTROGREEN1	UV_GOLDENROD1	UV_LIGHTOLIVE11	UV_REDYELLOW1
UV_BISTROGREEN2	UV_GOLDENROD2	UV_LIGHTOLIVE1 2	UV_REDYELLOW2
UV_BLAZE_ORANGE1	UV_GRASSGREEN1	UV_LIGHTPURPLE1	UV_ROSEPINK1
UV_BLAZE_ORANGE2	UV_GRASSGREEN2	UV_LIGHTPURPLE2	UV_ROSEPINK2
UV_BLUE1	UV_GRAYMAROON1	UV_LIMEGREEN1	UV_ROSERED1
UV_BLUE2	UV_GRAYMAROON2	UV_LIMEGREEN2	UV_ROSERED2
UV_BLUE11	UV_GREENERPASTURES1	UV_LUSTREBLUE1	UV_RUBY1
UV_BLUE12	UV_GREENERPASTURES2	UV_LUSTREBLUE2	UV_RUBY2
UV_BLUEBLUE1	UV_GREENYELLOW1	UV_MAJOLICABLUE1	UV_RUSTICBROWN1
UV_BLUEBLUE2	UV_GREENYELLOW2	UV_MAJOLICABLUE2	UV_RUSTICBROWN2
UV_BLUEGRAY	UV_GREENYELLOW11	UV_MAROON1	UV_SCREAMINGGREEN1
UV_DARKBLUE	UV_GREENYELLOW1 2	UV_MAROON2	UV_SCREAMINGGREEN2
UV_BLUEGRAY1	UV_HUNTERGREEN1	UV_MEDGRAY1	UV_SEAFOAMGREEN1
UV_BLUEGRAY2	UV_HUNTERGREEN2	UV_MEDGRAY2	UV_SEAFOAMGREEN2
UV_BRONZEMIST1	UV_JUNEBUG1	UV_MEDIUMSEAGREEN1	UV_SKYBLUE1
UV_BRONZEMIST2	UV_JUNEBUG2	UV_MEDIUMSEAGREEN2	UV_SKYBLUE2
UV_BUTTERNUT1	UV_JUNGLEGREEN1	UV_OLIVE1	UV_SPRINGGREEN1
UV_BUTTERNUT2	UV_JUNGLEGREEN2	UV_OLIVE2	UV_SPRINGGREEN2
UV_CHIPMUNK1	UV_LACQUERRED1	UV_ORANGEROD1	UV_STRAW1
UV_CHIPMUNK2	UV_LACQUERRED2	UV_ORANGEROD2	UV_STRAW2
UV_DARKBROWN1	UV_LAVENDER	UV_PACIFIC1	UV_TAUPEROD1
UV_DARKBROWN2	UV_TAN	UV_PACIFIC2	UV_TAUPEROD2
UV_DARKBROWN11	UV_LEAFGREEN1	UV_PALEBLUE1	UV_TURTLEGREEN1
UV_DARKBROWN1 2	UV_LEAFGREEN2	UV_PALEBLUE2	UV_TURTLEGREEN2
UV_DARKOLIVE	UV_LIGHTBLUE	UV_PARKNAVY1	UV_VERMILION1
UV_DARKPURPLE	UV_SEAFOAM	UV_PARKNAVY2	UV_VERMILION2
UV_DARKORANGE1	UV_LIGHTBROWN1	UV_PINEBARK1	UV_VETIVER1
UV_DARKORANGE2	UV_LIGHTBROWN2	UV_PINEBARK2	UV_VETIVER2
UV_DARKPINK	UV_LIGHTBROWN11	UV_PLUMPERFECT1	UV_YELLOW1

FLUORESCENT COLOR NAMES (NO COLOR MANAGEMENT SETTING REQUIRED.)			
UV_WILDROSE	UV_LIGHTBROWN12	UV_PLUMPERFECT2	UV_YELLOW2
UV_DARKSKY1	UV_LIGHTGRAY1	UV_PURPLEGRAY1	UV_YELLOWISHGREEN1
UV_DARKSKY2	UV_LIGHTGRAY2	UV_PURPLEGRAY2	UV_YELLOWISHGREEN2
UV_DEEPPINK1	UV_LIGHTGREEN	UV_REDDISHORANGE1	
UV_DEEPPINK2	UV_LIME	UV_REDDISHORANGE2	
UV_DEEPTAL1	UV_LIGHTGREENGRAY1		
UV_DEEPTAL2	UV_LIGHTGREENGRAY2		

FLUORESCENT COLOR NAMES (DIRECT CMYK COLOR MANAGEMENT REQUIRED.)			
UV_BROWN1	UV_GOLD1	UV_PINK	UV_PURPLE1
UV_BROWN2	UV_GOLD2	UV_SALMON	UV_PURPLE2
UV_DARKGRAY1	UV_GREEN1	UV_PURPLE	UV_RUSSIANBLUE
UV_DARKGRAY2	UV_GREEN2	UV_VANITY	UV_SAGE
UV_DARKPINK1	UV_LIGHTOLIVE1	UV_REDDISHYELLOW1	
UV_DARKPINK2	UV_LIGHTOLIVE2	UV_REDDISHYELLOW2	

When creating a Two-Layer Fluorescent text effect, the visible text (top layer) must use one of these color keys:

VISIBLE UV TEXT COLOR
UV_2L_CYAN
UV_2L_MAGENTA
UV_2L_YELLOW

The table that follows is a guide that shows what two layer color works best with the underlying fluorescent color.

FLUORESCENT TWO-LAYER COLOR CHART	
BOTTOM LAYER UV_COLOR	TOP LAYER FLUORESCENT COLOR (THE VISIBLE TEXT)
UV_BLUEBLUE1 (deprecated)	UV_2L_MAGENTA or UV_2L_YELLOW
UV_BISTROGREEN1	UV_CYAN or UV_MAGENTA or UV_YELLOW
UV_BISTROGREEN2	UV_CYAN or UV_MAGENTA or UV_YELLOW
UV_BLAZE_ORANGE1	UV_CYAN or UV_YELLOW
UV_BLUEBLUE2 (deprecated)	UV_2L_MAGENTA or UV_2L_YELLOW
UV_BRONZEMIST2	UV_CYAN or UV_YELLOW

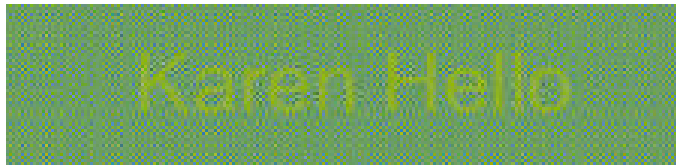
FLUORESCENT TWO-LAYER COLOR CHART	
BOTTOM LAYER UV_COLOR	TOP LAYER FLUORESCENT COLOR (THE VISIBLE TEXT)
UV_BUTTERNUT1	UV_CYAN or UV_YELLOW
UV_CHIPMUNK2	UV_CYAN or UV_YELLOW
UV_DARKBROWN11	UV_CYAN or UV_YELLOW
UV_DARKBROWN12	UV_CYAN or UV_YELLOW
UV_DARKORANGE1 (deprecated)	UV_2L_CYAN or UV_2L_MAGENTA or UV_2L_YELLOW
UV_DARKPINK1	UV_2L_CYAN
UV_DARKSKY1 (deprecated)	UV_2L_YELLOW
UV_GOLDENROD1 (deprecated)	UV_2L_CYAN or UV_2L_MAGENTA or UV_2L_YELLOW
UV_GRASSGREEN1 (deprecated)	UV_2L_CYAN or UV_2L_MAGENTA
UV_GREENERPASTURES1	UV_CYAN or UV_MAGENTA or UV_YELLOW
UV_GREENERPASTURES2	UV_CYAN or UV_MAGENTA or UV_YELLOW
UV_HUNTERGREEN1	UV_CYAN or UV_YELLOW
UV_HUNTERGREEN2	UV_CYAN or UV_MAGENTA or UV_YELLOW
UV_JUNEBUG2	UV_CYAN or UV_MAGENTA or UV_YELLOW
UV_JUNGLEGREEN1	UV_CYAN or UV_MAGENTA or UV_YELLOW
UV_JUNGLEGREEN2	UV_CYAN or UV_MAGENTA or UV_YELLOW
UV_LEAFGREEN1	UV_2L_CYAN or UV_2L_MAGENTA
UV_LEAFGREEN2	UV_2L_MAGENTA or UV_2L_YELLOW
UV_LIMEGREEN1	UV_2L_CYAN or UV_2L_MAGENTA or UV_2L_YELLOW
UV_MAROON1 (deprecated)	UV_2L_CYAN or UV_2L_YELLOW
UV_MEDIUMSEAGREEN1	UV_2L_MAGENTA
UV_ORANGEROD1 (deprecated)	UV_CYAN or UV_MAGENTA
UV_PACIFIC2 (deprecated)	UV_CYAN or UV_YELLOW
UV_PARKNAVY1 (deprecated)	UV_CYAN or UV_YELLOW
UV_PARKNAVY2 (deprecated)	UV_CYAN or UV_YELLOW
UV_PINEBARK2 (deprecated)	UV_CYAN
UV_PLUMPERFECT1 (deprecated)	UV_CYAN or UV_YELLOW

FLUORESCENT TWO-LAYER COLOR CHART	
BOTTOM LAYER UV_COLOR	TOP LAYER FLUORESCENT COLOR (THE VISIBLE TEXT)
UV_PLUMPERFECT2 (deprecated)	UV_CYAN or UV_YELLOW
UV_REDDISHORANGE1	UV_2L_CYAN or UV_2L_MAGENTA
UV_REDDISHORANGE2	UV_2L_CYAN or UV_2L_MAGENTA
UV_ROSERED1	UV_2L_CYAN
UV_SPRINGGREEN1	UV_2L_YELLOW
UV_SPRINGGREEN2	UV_2L_YELLOW
UV_STRAW1 (deprecated)	UV_2L_MAGENTA
UV_STRAW2 (deprecated)	UV_2L_CYAN or UV_2L_MAGENTA or UV_2L_YELLOW
UV_TAUPEROD2 (deprecated)	UV_CYAN or UV_YELLOW
UV_TURTLEGREEN1 (deprecated)	UV_CYAN or UV_MAGENTA or UV_YELLOW
UV_TURTLEGREEN2 (deprecated)	UV_CYAN or UV_MAGENTA or UV_YELLOW
UV_VERMILION1	UV_2L_CYAN UV_2L_MAGENTA
UV_VERMILION2	UV_2L_MAGENTA
UV_VETIVER1 (deprecated)	UV_CYAN or UV_MAGENTA or UV_YELLOW
UV_YELLOW1	UV_2L_YELLOW
UV_YELLOWISHGREEN1 (deprecated)	UV_CYAN or UV_MAGENTA or UV_YELLOW
UV_YELLOWISHGREEN2 (deprecated)	UV_2L_YELLOW

## DESIGNING DOCUMENTS WITH FLUORESCENTMARK COLORS

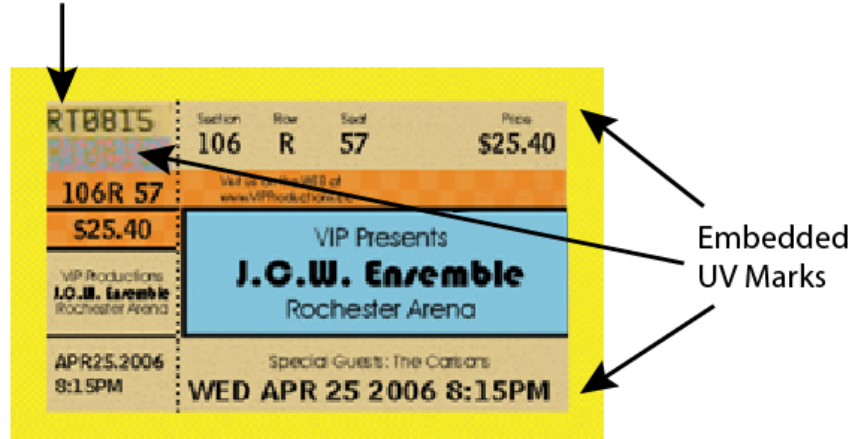
The method by which the FluorescentMark colors are created and rendered makes it difficult to alter electronically or physically, and if designed well, makes it difficult for a user to even realize that information is embedded in textured color patterns. Another benefit of FluorescentMark colors is that to decode the hidden information, only a UV light source is required. Many inexpensive types of UV sources exist (e.g., portable money detectors, flashlights, and even key chains).

FluorescentMark colors are used to embed a text string into a textured color pattern in such a way that the text is indecipherable under normal viewing light, and visible only under ultra-violet (UV) lighting. In addition, a second layer of visible text can be used to further hide or restrict the visibility of the hidden layer. When viewed under UV illumination the top layer of text will fade to allow the hidden layer to become visible.

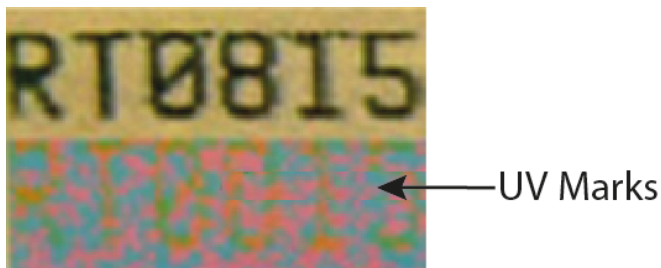


In the concert ticket example below a unique ticket identification (ID) number is included on the ticket. The same ticket ID can be encoded as FluorescentMark colors, which appear as textured color patterns at different locations on the ticket.

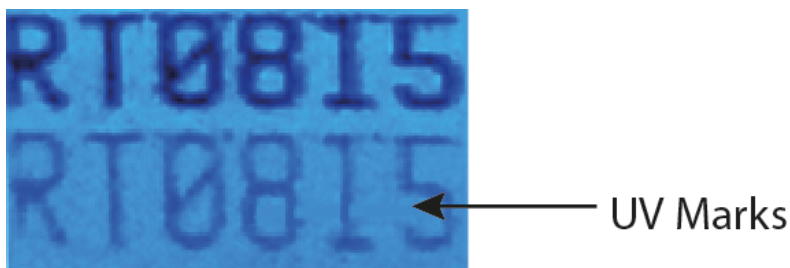
Ticket ID



In the example below, a portion of the ticket is enlarged to show the ticket ID in plain text with FluorescentMark colors immediately below. The FluorescentMark color appears as a textured color pattern.



Authenticity of the ticket is established by viewing it under UV light and comparing the FluorescentMark colors with the visible ID number, as shown below. Parity indicates that it is an authentic ticket.



The same ticket portion from the previous illustration but viewed under UV light. The FluorescentMark colors now reveal a number that matches the visible ID.

Similar applications for FluorescentMark colors are on coupons, checks, college grade reports, etc.

## How FluorescentMark colors work

FluorescentMark colors are implemented within VIPP® using a Postscript construct called Pattern Inks. This is essentially a Postscript command for creating a "textured fill pattern" comprising a mosaic of one or more CMYK colors. Pattern inks come in pairs - a base pattern for the background, and a pattern for the text. The user specifies a region in the document where the FluorescentMark colors should be placed, selects a base pattern ink from the set of supported pattern inks and enters the text string to be embedded as FluorescentMark colors. The specified region is first filled with the base pattern ink. The text string is then painted inside the region with the corresponding text pattern ink.

## Guidelines for creating effective FluorescentMark color applications

<b>Printer</b>	FluorescentMark colors work only with supported color printers.
<b>Paper stock</b>	A fundamental requirement is that the paper is of high brilliance and thus contains brightening agents, so that the text fluoresces under UV light. Most papers used in digital color printing have high "Brilliance" numbers. The higher the brilliance number, the stronger the effect, do not use dull or colored paper.
<b>Color calibration</b>	The printer must be color calibrated. A quick way to check if a printer is within calibration is to print the FluorescentMark colors swatch book, and examine the two rows of gray patches at the bottom of the first sheet, see the example below. The top row of gray patches must approximately match the bottom row of gray patches through most of the ramp, except possibly near black. When the two rows appear vastly different, the printer must be calibrated. Use the FreeFlow Print Server built-in calibration tool. It is strongly recommended that the same paper stock and queue that will be used to print the final document is also used in the calibration procedure.
<b>Fonts</b>	The font, size and typeface of the text string used to encode the FluorescentMark colors are chosen by the designer. The pattern inks have been designed to work best for font sizes between 10 and 14 points. Sans serif fonts such as Helvetica, Arial, or Geneva are recommended. Serif fonts, highly stylized fonts and special symbols should be avoided. Xerox recommends that the designer experiment with different fonts and typefaces to achieve the best trade-off between the two competing factors of: making the text indecipherable under normal light, and making the text readily decipherable under UV light.



Note: Normal light may also contain a UV component, which may increase the visibility of FluorescentMark colors, which is particularly true of daylight. FluorescentMark colors may be visible under normal light as a gloss effect when the document is viewed at an angle. Fonts should be selected to minimize these effects.

<b>Document Design</b>	It is generally recommended that FluorescentMark colors are included in a portion of the document that does not ordinarily draw much attention. Good locations are the borders of the document. Xerox recommends the text string be kept short, and replicated many times within the pattern. Replication facilitates rapid detection of the FluorescentMark color under UV light.
------------------------	--



Note: A layer of text visible under normal lighting conditions can be placed on top of the fluorescent area. Ensure that the text uses one of the special top layer colors. The top layer of text fades to allow the underlying text to show through when a UV illumination source is used.

## TWO-LAYER FLUORESCENT TEXT

Two-layer for Fluorescent text uses an additional layer of visible text over an existing fluorescent effect. The visible layer of text will fade under UV illumination revealing the text in the Fluorescent effect.

Ensure that the top layer of text (the visible text) is assigned one of three color values (GEPkey), CYAN, MAGENTA or YELLOW, using the **SETTXXC** command. This command has been extended to support an additional parameter UV2L, which when used indicates this will be used in a two-layer Fluorescent effect:

```
POINT SETUNIT
40 680 MOVETO
{
/NHEB 20 SETFONT
140 -10 375 -30 UV_DARKPINK1 DRAWB           % Sets the UV color for box and hidden text
148 -3 MOVETO (Bottom Layer Text - Hidden) SH
[CYAN UV2L ] SETTXXC                          % Sets the top layer color
148 -3 MOVETO
(This is the top layer visible Text) SH
} BCALL
```

## Design Considerations for GlossMark® Text

You find the following Information here:

- [GlossMark Text limitations and requirements](#)
- [GlossMark Colorkey table](#)
- [Designing documents with GlossMark Text](#)

The differential gloss characteristics of GlossMark Text make it difficult to alter electronically or physically. When designed well, the GlossMark Text also has an aesthetic component, increasing the perceived value of the ticket.

These two figures show GlossMark Text produced on standard coated paper. The first image is in straight-on view. GlossMark Text works best with constant colors, but color gradations can also be used. VIPP® supplies 13 pre-defined Colorkeys (listed in [GlossMark Colorkey table](#)), or define individual Colorkeys as necessary.



GlossMark Text available on (Xerox only) FFPS and EFI Full Color only, refers to printing variable text strings in a way that the content is not visible in straight-on view, but becomes visible as gloss under inclined illumination.

Once defined with the SETPAT command GlossMark effect can be printed using any font available on your printer.

### Note on legacy GlossMark fonts

The legacy method of printing GlossMark text using a collection of dedicated fonts is deprecated. It is still available and supported for backward compatibility but for new applications it is recommended to use the new GlossMark feature which offers more flexibility.

A GlossMark effect is defined using a SETPAT statement with the following syntax (refer to the VIPP® Language Reference Manual for syntax details):

```
/PATkey [ /VPGL Iwidth coll col2 ] Htr Vtr rot Hscale Yscale SETPAT
```

## GLOSSMARK TEXT LIMITATIONS AND REQUIREMENTS

The media stock for GlossMark Text has to be smooth and the font color should be selected from a given palette.

For a more complete list of limitations refer to the [Limitations](#) section of this document.

<b>Paper Requirements</b>	GlossMark Text requires paper with a smooth glossy surface, for example, a coated glossy paper.
<b>GlossMark Text Color</b>	VI Compose supports 13 named colors.
<b>Text Content</b>	Arbitrary text content, the text string should be reasonably short, since color shifts can occur in large constant areas. By repeating information you increase the readability of the text.
<b>DFEs</b>	FFPS and EFI.

## GLOSSMARK COLORKEY TABLE

The available colors for GlossMark effects are prefixed with GL\_ and are listed in the table below.

COLORKEYS			
GL_Black	GL_Green	GL_Margenta	GL_Peach
GL_Blue	GL_lightBlue	GL_Maroon	GL_Red
GL_CyanGL_Gray	GL_lightGreen	GL_Olive	GL_Yellow



Note: Do not change the name or the color settings on these color definitions. The color names listed give an approximation of the actual color pattern that will be printed on the print device.

## DESIGNING DOCUMENTS WITH GLOSSMARK TEXT

The differential gloss characteristics of GlossMark Text make it difficult to reproduce or alter electronically or physically. If designed well, the GlossMark Text also has an aesthetic component, increasing the perceived value of the ticket.

GlossMark Text is printed in a constant color that, under certain angles of illumination, will show a variable data string in the local gloss of the solid color.

In the concert ticket example below, the constant color area below the ticket price can be used to hold variable data.



In the example below, the GlossMark Text area is enlarged in "straight-on" view, showing a solid color.



In the example below, the same area is shown under an angled illumination, revealing the ticket price as gloss variation.



Similar applications for GlossMark Text are coupons, checks, college grade reports, etc.

#### **Guidelines for creating effective GlossMark Text applications**

Keep these points in mind when designing applications using GlossMark Text:

<b>Printer</b>	GlossMark Text is supported on Xerox color devices using FFPS or EFI DFEs. (Low Gloss EA Toner will greatly reduce the effectiveness of this effect.)
<b>Media</b>	A fundamental requirement is that the media surface is extremely smooth. Coated papers - preferably gloss coated - papers work well. Plastics, never-tear and similar polymers also work well.
<b>GlossMark Colors</b>	Refer to <a href="#">GlossMark Colorkey table</a> .

## Design Considerations for Infrared Color

Infrared color, available on Xerox only, FFPS and EFI Full Color engines, refers to printing variable strings in a way that the content is not visible under normal light, but you can view when you use Infrared light. Predefined Infrared color keys are listed in a table at the end of this section.

### INFRARED COLORKEYS

Information found here includes:

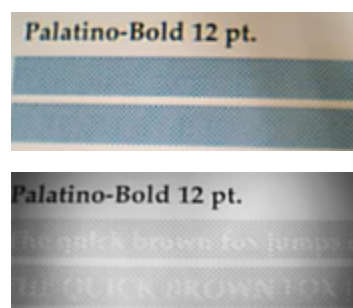
- [Infrared Colorkey tables](#)
- [Designing Documents with Infrared color](#)

Infrared effects are available as a single layer or two-layer effects. A single layer effect hides a string of text under a solid color, while a two-layer infrared effect hides a string of text under a color that has additional text printed on top. When an infrared light source is used to view the effects, the top layer of text (two-layer) and the area of color covering the hidden text will disappear revealing the bottom layer of text.

### DESIGNING DOCUMENTS WITH INFRARED COLOR

The method by which the Infrared color application is created and rendered makes it difficult to reproduce or alter electronically or physically, and when designed well, makes it difficult for a user to even realize that information is embedded in textured color patterns. Another benefit of Infrared colors is that the decoding of the hidden information can be accomplished with a standard web-camera fitted with IR illumination for "night vision". Many inexpensive webcam models exist, which will illuminate the object with IR light, capture the resulting reflected light, and digitally convert the raw captured video into a grayscale video stream viewable on a standard computer.

These two figures show a color patch under normal illumination (top) and under Infrared illumination (bottom). The illustration on the bottom was captured using a web cam and displayed on a PC.



### INFRARED COLORKEY TABLES

SINGLE LAYER IR COLORS THAT DO NOT REQUIRE DIRECT CMYK COLOR MANAGEMENT			
IR_ARGYLEPINK1	IR_DARKRED1	IR_JASMINEGREEN1	IR_ORANGERED21
IR_ARGYLEPINK2	IR_DARKRED2	IR_JASMINEGREEN2	IR_ORANGERED22
IR_BLACKSLATE1	IR_DARKTEAL1	IR_KNIT1	IR_PURPLE1
IR_BLACKSLATE2	IR_DARKTEAL2	IR_KNIT2	IR_PURPLE2

SINGLE LAYER IR COLORS THAT DO NOT REQUIRE DIRECT CMYK COLOR MANAGEMENT			
IR_BLUEPURPLE1	IR_DARKYELLOW1	IR_LAWNGREEN1	IR_RASPERBERRY1
IR_BLUEPURPLE2	IR_DARKYELLOW2	IR_LAWNGREEN2	IR_RASPERBERRY2
IR_BROWN1	IR_DEEPPURPLE1	IR_LIGHTBROWN1	IR_RED1
IR_BROWN2	IR_DEEPPURPLE2	IR_LIGHTBROWN2	IR_RED2
IR_BRONZEGREEN1	IR_DEEPRED1	IR_LIGHTPINK1	IR_ROSEVIOLET1
IR_BRONZEGREEN2	IR_DEEPRED2	IR_LIGHTPINK2	IR_ROSEVIOLET2
IR_CINNABAR1	IR_EVERGREEN1	IR_LIGHTPURPLE1	IR_RHODODENDRON1
IR_CINNABAR2	IR_EVERGREEN2	IR_LIGHTPURPLE2	IR_RHODODENDRON2
IR_COCONUTSHELL1	IR_FIREBRICK1	IR_MALLARDGREEN1	IR_ROSEPINK1
IR_COCONUTSHELL2	IR_FIREBRICK2	IR_MALLARDGREEN2	IR_ROSEPINK2
IR_CORDOVAN1	IR_GRAPEJUICE1	IR_MAZARINE1	IR_SNORKELBLUE1
IR_CORDOVAN2	IR_GRAPEJUICE2	IR_MAZARINE2	IR_SNORKELBLUE2
IR_CRIMSON1	IR_GRAYMAROON1	IR_MAROON1	IR_TEAL1
IR_CRIMSON2	IR_GRAYMAROON2	IR_MAROON2	IR_TEAL2
IR_DARKGREEN21	IR_GRAYGREEN1	IR_MAROON21	IR_VIOLETKNIT1
IR_DARKGREEN22	IR_GRAYGREEN2	IR_MAROON22	IR_VIOLETKNIT2
IR_DARKPINK1	IR_GREEN1	IR_OLIVE1	IR_YELLOW1
IR_DARKPINK2	IR_GREEN2	IR_OLIVE2	IR_YELLOW2
IR_DARKPURPLE1	IR_GREENYELLOW1	IR_OLIVE21	IR_YELLOWGREEN1
IR_DARKPURPLE2	IR_GREENYELLOW2	IR_OLIVE22	IR_YELLOWGREEN2
IR_DARKORANGE1	IR_GOLDENROD1	IR_ORANGERED1	
IR_DARKORANGE2	IR_GOLDENROD2	IR_ORANGERED2	

SINGLE LAYER IR COLORS THAT REQUIRE DIRECT CMYK COLOR MANAGEMENT			
IR_BLUE1	IR_DEEPTOAL1	IR_DP_LIGHTORANGE1	IR_DP_PINK1
IR_BLUE2	IR_DEEPTOAL2	IR_DP_LIGHTORANGE2	IR_DP_PINK2
IR_DARKBLUE1	IR_DEEPBLUE1	IR_DP_LIGHTYELLOW1	IR_DP_PURPLE1
IR_DARKBLUE2	IR_DEEPBLUE2	IR_DP_LIGHTYELLOW2	IR_DP_PURPLE2
IR_DARKGREEN1	IR_DP_CYAN1	IR_DP_MAGENTA1	IR_DP_SUBLIME1
IR_DARKGREEN2	IR_DP_CYAN2	IR_DP_MAGENTA2	IR_DP_SUBLIME2

SINGLE LAYER IR COLORS THAT REQUIRE DIRECT CMYK COLOR MANAGEMENT			
IR_DARKOLIVE1	IR_DP_FIREBRICK1	IR_DP_ORANGE1	IR_DP_YELLOW1
IR_DARKOLIVE2	IR_DP_FIREBRICK2	IR_DP_ORANGE2	IR_DP_YELLOW2

TWO-LAYER IR COLORS THAT REQUIRE DIRECT CMYK COLOR MANAGEMENT			
IR_2L_BLUEPURPLE1	IR_2L_LIGHTCYAN1	IR_2L_LIGHTOLIVE1	IR_2L_ORANGE1
IR_2L_BLUEPURPLE2	IR_2L_LIGHTCYAN2	IR_2L_LIGHTOLIVE2	IR_2L_ORANGE2
IR_2L_DARKPINK1	IR_2L_LIGHTGRAY1	IR_2L_LIGHTPURPLE1	IR_2L_PALEBLUE1
IR_2L_DARKPINK2	IR_2L_LIGHTGRAY2	IR_2L_LIGHTPURPLE2	IR_2L_PALEBLUE2
IR_2L_GRAYBROWN1	IR_2L_LIGHTGRAY21	IR_2L_LIGHTPURPLE21	IR_2L_PURPLE1
IR_2L_GRAYBROWN2	IR_2L_LIGHTGRAY22	IR_2L_LIGHTPURPLE22	IR_2L_PURPLE2
IR_2L_GREENYELLOW1	IR_2L_LIGHTGREEN1	IR_2L_LIGHTRED1	
IR_2L_GREENYELLOW2	IR_2L_LIGHTGREEN2	IR_2L_LIGHTRED2	
IR_2L_LIGHTBROWN1	IR_2L_LIGHTMAGENTA1	IR_2L_LIGHTYELLOW1	
IR_2L_LIGHTBROWN2	IR_2L_LIGHTMAGENTA2	IR_2L_LIGHTYELLOW2	

When creating a Two-Layer Infrared text effect, the visible text (top layer) must use one of these color keys:

VISIBLE IR TEXT COLOR
IR_Black
IR_Blue
IR_Green
IR_Red

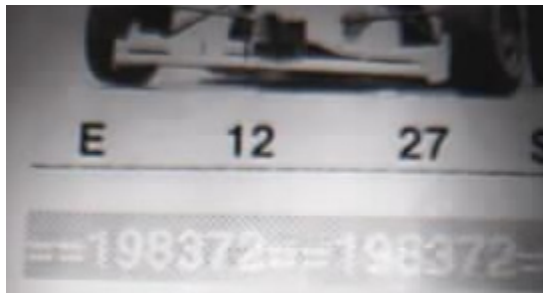
## DESIGNING DOCUMENTS WITH INFRARED COLOR

Infrared colors are used to embed a text string into a textured color pattern in such a way that the text is hidden under normal viewing light, and visible only under infrared (IR) illumination. It is quite similar to FluorescentMark color technology, the main difference is that IR illumination is required instead of UV illumination in order to reveal the hidden text.

In a Two-layer Infrared color application an extra layer of information using Overprint can be placed on top of the existing IR text. The overprinted text uses solid colors and is visible under normal illumination but will disappear under IR illumination revealing the underlying IR text. This has the advantage of being able to add another set of variable data as well as extra level of distraction to hide the IR data.

In a Two-layer Infrared color application the designer can add another variable string of text as an overprint. The examples below are a race ticket with a solid area where the ticket number verifier is encoded with Infrared colors. A

variable text that is not security sensitive (in this example the Section, Row, and Seat numbers) is printed as overprint solid color text. Under normal illumination only the overprint text is visible. Under IR illumination the overprint text is not visible anymore and the Infrared color encoded text is revealed.



Also, note that Text is only one type of object that can be used with Infrared colors. There can be other applications such as inserting graphics or logos. As long as the right color set is used any type of information can be embedded as an Infrared color.

#### Guidelines for creating effective Infrared color applications

<b>Printer</b>	Infrared colors are supported on Xerox color devices using FFPS or EFI DFEs.
<b>Choice of paper stock</b>	Infrared colors should work well on any standard paper stock. Xerox recommends testing the job before a production run.
<b>DFE settings</b>	Infrared color has been tested extensively with the FreeFlow Print Server. The image path or queue should be set for printing raw device CMYK; i.e., there should be NO press emulation or any other mode that will transform the CMYK colors in the document. The only function that should be in the image path is color calibration.
<b>Colors</b>	Depending on the printer setup, some ink pairs will provide stronger IR contrast than others. Note also that each ink is really a pair, marked by 1 and 2. Both inks have similar appearance under normal light. However under IR light, ink 1 is always darker than ink 2. Therefore the use of ink 1 as the background will result in light text against dark background under IR light, and conversely the use of 2 as the background will result in dark text against light background under IR.

For Two-layer applications the set of recommended Solid Colors to use for the overprint is currently: Black (CMY Black), Blue, Red, and Green. Other solid colors can also work well depending on the IR background color choices.

**Choice of fonts**

The font, size and typeface of the text string used to encode the Infrared colors are optional. The supplied pattern inks have been designed to work best for font sizes between 10 and 12 points. Sans serif fonts, for example, Helvetica, Arial, Geneva are recommended. Serif fonts, highly stylized fonts and special symbols should be avoided. Xerox recommends that the designer experiment with different fonts and typefaces to achieve the best trade-off between the two competing objectives (hiding the text under normal light, and making the text readily decipherable under IR light). Also, Infrared colors may be visible under normal light as a gloss effect when the document is viewed at an angle. As a general rule, Infrared colors producing stronger contrast under IR light may also have increased visibility under normal light. In these cases, smaller or plain-face fonts may be more suitable. Conversely, Infrared colors that hide very well under normal light may also be more difficult to discern under IR light. For these cases, larger and/or boldface fonts may be more appropriate.

**Document design**

It is generally recommended that Infrared colors are included in a portion of the document that does not ordinarily draw much attention. Good locations are the borders of a document. Xerox recommends that the text string be kept short, and replicated many times within the pattern. Replication facilitates robust and reliable detection of the Infrared colors under IR light.

## Design Considerations for MicroText Font

Information found here includes:

- [MicroText font limitations and requirements](#)
- [MicroText font table](#)
- [Designing documents with MicroText fonts](#)

The advantages of MicroText font applications stem from the font size, which makes the output difficult to alter electronically or physically. When designed well, the size of the output makes it difficult for a user to even realize that information is embedded in the document. A simple loupe or magnifier can be used to make the MicroText font visible.

The top figure shows the quality of a MicroText font printed on standard coated paper. For comparison, in the bottom figure, the microprinting on Japanese currency is shown at identical scale. When judging quality, it is useful to compare the results to currency printing.



MicroText fonts can be used on Xerox FFPS and EFI devices. (Not supported on Highlight Color). MicroText refers to printing text strings at a point size smaller than 1 pt. Typically information in a string is repeated to avoid loss or readability due to fibers in the paper making a character unreadable. Using the Repeat Data Transform feature in VI Design Express is an easy way to repeat a text string.



Note: If necessary, increase the view magnification of the document to see the MicroText.

MicroText requires a smooth media to print on. Excessive fiber in the paper can make some micro characters unreadable, this is why the repetition of text items is recommended.

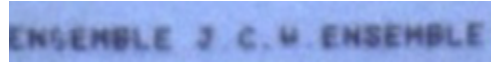
As with all Specialty Imaging applications, test the application to validate the effect meets the design specifications and print performance.

### MICROTEXT FONT LIMITATIONS AND REQUIREMENTS

Text applications using MicroText font are implemented within VIPP® using a Postscript font. This font is not scalable (different sizes are actually created by using different fonts) and you cannot rotate, except for integer multiples of 90°. The MicroText font supports the most common character set but is restricted to Upper Case

letters, numbers and few symbols. Special characters, such as ß, Ä, Š, ©, Æ, and the like, are not included in the set. Lower case characters are mapped to upper case characters for printing purposes.

The high resolution of MicroText font applications is directly impacted by the paper quality (coating) as can be seen below. The figure below shows the same string depicted in the previous example, this time printed on a lower quality paper.



It is important to know that in a MicroText font application, not all characters might be individually readable at all locations (refer to "S" in the first "ENSEMBLE") and that non-redundant text (license plates, user IDs, etc.) should be repeated.

For a more complete list of limitations refer to the [Limitations](#) section of this document.

<b>Paper Requirements</b>	Rough papers will increase defects. The Xerox Digital Color Elite Gloss families of paper products are recommended.
<b>Text Color</b>	Any solid color can be selected as the font color. Black, Cyan, Magenta and Yellow are recommended. Black is the only color supported on Monochrome HLC devices.
<b>Text Content</b>	Text should be repeated or redundant. Text containing MicroText font should not be used for the single occurrence of any non-redundant item, such as license plate or credit card numbers. By repeating information the readability of the text is improved.
<b>Font</b>	Upper-case, limited character set. Contains only these characters lower case characters have been mapped to upper case for printing purposes: <b>A through Z, 0 through 9, ! @ # \$ % &amp; * ( ) - _ = + [ ] ; : ' " , . ? / € , and ¥.</b> , and Cyrillic characters.
<b>DFEs</b>	FFPS and EFI. The high data content of text consisting of MicroText font can lead to a DFE slow-down.
<b>Color</b>	The patch color is defined as DEVICE CMYK, no emulation (SWOP, Fogra, EuroScale, etc.) can be performed. Solid separations only.
<b>Scale</b>	Any change in scale will destroy the font and is strongly discouraged.
<b>Rotation</b>	Any rotation, other than multiples of <b>90°</b> , has a high likelihood of destroying the font. Rotation is strongly discouraged.

## MICROTEXT FONT TABLE

The available MicroText fonts are:

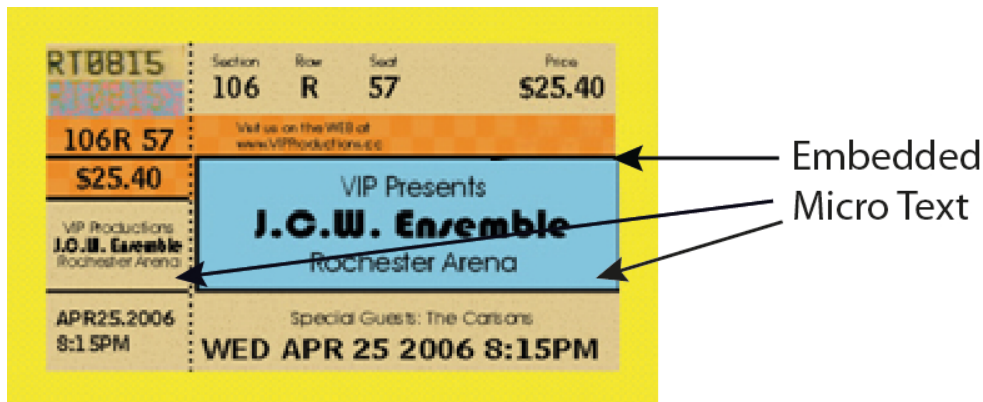
FONT NAMES	FONT STYLE
micro	f6, f6-Bold, f6-5, f6-5-Bold, f7, f7-Bold, f9, f9-Bold

## DESIGNING DOCUMENTS WITH MICRTEXT FONTS

MicroText font text refers to the printing of variable data strings using font sizes below 1 point. The size of MicroText fonts makes them difficult to reproduce or alter electronically or physically, and if designed well, makes it difficult for a user to even realize that information is embedded in the document. A simple loupe or magnifier can be used to make the MicroText font visible.

Below, a portion of the ticket is enlarged to show the MicroText font on the ticket:

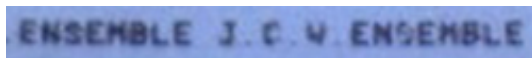
In the concert ticket example below, several unique text / and symbol strings are present on the ticket. MicroText font can be used to repeat these unique identifiers in a non-obvious way. One approach is to replace or augment the lines in the ticket design with this variable data. It is important to note, that conventional microprinting might have a fixed string as part of the design, but that VIPP® MicroText fonts are capable of personalizing the microprinting with the unique ticket identifiers.



Below, a portion of the ticket is enlarged to show the MicroText font on the ticket:



Below, a different line in the ticket design is enlarged to show a different part of the ticket ID printed in MicroText font.

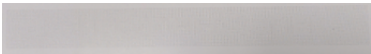



Applications using MicroText font can include coupons, checks, college grade reports, and so on.

## Design Considerations for Void Pantograph

The characteristics of a void pantograph make it useful to use to add security features to a document.

The figures show an example of a void pantograph text application. The first image is the original print and the foreground text is hidden. The second image is the duplicate, and the foreground text is visible:

Original	Duplicate
	

Void pantograph works best with constant colors, such as a combination of black and white, on coated or uncoated media. The `SI_VP_VoidPanto.vpc` is a printable demo file for the void pantograph effect. You can find this file in the `xgf/demo` folder. The void pantograph text feature allows you to print variable text strings in a way that the content is not visible unless the text from the original print is duplicated. After you define the void pantograph using the **SETPAT** command, to print the void pantograph effect, use any font available on your printer.



Note: The **SHX** command rotation does not produce good results with void pantograph. The color or pattern of the letters override the effect.

### Note on legacy fonts

Void pantograph is available as a text effect. To define this effect, use a **SETPAT** statement with the following syntax. For details, refer to the *VIPP® Language Reference Manual*:

```
/PATkey [/VPPG1 textPixelFrequency backgroundPixelFrequency color1 color2] Htrans
Vtrans rotate Hscale Yscale SETPAT
```

PATKEY	Pattern key. The user can provide a name for the pattern for example, /VP01.
/VPPG1	Identifier for the void pantograph pattern for <b>SETPAT</b> command. Do not change this identifier.
textPixelFrequency	Number range, 9–20. This number provides the text pixel frequency.
backgroundPixelFrequency	Number range, 29–39. This number provides the pixel frequency of the background.
Color1	Color of the pixel and text background. It is recommended that you use <i>OWHITE</i> .
Color2	Color of the pixel. It is recommended that you use <i>BLACK</i> .
Htrans	Horizontal transformation
Vtrans	Vertical Transformation
Rotate	Rotation of pixels
Hscale	Horizontal scaling
Vscale	V scaling

You can print the void pantograph effect anywhere on the document. The best results are achieved when the effect is printed on a white background. Therefore, it is recommended to use the *OWHITE* color as the background color in

the void pantograph effect definition color1 argument of the **SETPAT**.

# VI Compose Services

This chapter contains:

Normalization ..... 250

Demographics ..... 252

File samples ..... 255

ACCLOG report 2 ..... 258

VI Compose Services include:

- Normalization
- Demographics

## Normalization

The normalization feature transforms VIPP® jobs so that they can be pre-processed before they are printed or imaged. For example, after normalization, a print server can split a job into segments and route each segment to a different printer. On FreeFlow Print Server DFEs, normalization enables parallel processing of the job's pages.

The normalization process converts most but not all VIPP® jobs into normalized VIPP® jobs. A normalized VIPP® job is a VIPP® native mode job with standard DSC for page delimitation (% % Page:) and a guarantee of page independence for each page.

Imaging is disabled by default during normalization.

Normalization is enabled by placing a statement as defined in Syntax at the beginning of:

- The submitted job
- A JDT, DBM, and XJT
- A VI Compose configuration file (xgf.def)

### Specify SETPARAMS in only one of these files

The SETPARAMS command should be specified in only one of these files to ensure the creation of only one target\_name file (see definition below) in either the first folder specified by SETMPATH or the local project folder. For FreeFlow Print Server, the target\_name file is always written to the folder that begins with /opt/XRXnps in SETMPATH.

### Where is the target\_name file?

When the SETPARAMS command precedes a SETPROJECT command, the target\_name file will be created in the first folder specified by SETMPATH. When the SETPARAMS command follows a SETPROJECT command, the target\_name file will be written to the local project folder. For FreeFlow Print Server, the target\_name file is always written to the folder that begins with /opt/XRXnps in SETMPATH.

### Warning raised

When no pages are produced this warning will be raised:

WARNING: VSM#3 Normalization produced empty file



**Warning:** Not all VIPP® jobs can be normalized. Especially VIPP® jobs with PostScript sequences not encapsulated by SETFORM or SCALL. These will fail to normalize.

For example:

- Hybrid mode jobs (emitted by some VDP applications) cannot be normalized.
- Jobs including multi-page PostScript files using the RUN command cannot be normalized.
- Jobs using the VIPP® ZSORT command.

### Syntax

```
[ /Normalize [ option (target_name) ] ] SETPARAMS
```

### Where:

- Option may be:

- **0** do not normalize
- **1** normalize without imaging
- **101** normalize without imaging, and mark normalized pages with a circled *N* at the bottom left corner to ease the comparison of original to normalized pages.
- **1001** normalize with imaging
- **1101** normalize with imaging, and mark normalized pages with a circled "N" at the bottom left corner to ease the comparison of original to normalized pages.
- **2** normalize without imaging, emit % %Page DSC for front pages only
- **102** normalize without imaging, emit % %Page DSC for front pages only and mark normalized pages with a circled "N".
- **1002** normalize with imaging, emit % %Page DSC for front pages only.
- **1102** normalize with imaging, emit % %Page DSC for front pages only and mark normalized pages with a circled *N*.
- Target name is the name of the file that will receive the normalized job. The extension .vpn will be automatically appended to the file name. If the job is run on VIeC in multi-instances mode a time stamp will also be appended to target\_name to ensure file name uniqueness if several jobs using the same target\_name are processed by VIeC in parallel. The file created in one of these folder is:
  - the first writable folder specified by SETMPATH in xgfunix.run, or the project folder in project mode
  - for FreeFlow Print Server, the folder that begins with /opt/XRXnps in SETMPATH

## Demographics

The job demographics functionality captures information about a VIPP® job. The information captured includes the number of:

- Pages generated
- Copies generated
- Sets generated
- Pages generated per media type
- Images, forms, jdt's, fonts, and miscellaneous resources

Demographics tracks the job-related information generated, and resources used in a job. The information can be used for job-estimation, in accounting files, and with integrity checking systems of output devices.



Note: When the RUN command is used on multi page PostScript files it must be called with options 2 or 3 to provide accurate figures regarding page count.

The /Demographics parameter replaces /ChkResources in SETPARAMS, however, /ChkResources is supported for backward compatibility. When running the demographics service, /ChkResources statements are automatically converted to /Demographics statements. As a result:

- The vipp.vsm module must be active for /ChkResources statements to work
- /ChkResources statements in the existing code should be replaced with /Demographics statements

This is a sample of a /ChkResources statement before and after conversion:

- [ /ChkResources PD ] SETPARAMS
- [ /Demographics [ 011PD (reslist.dbf) ] ] SETPARAMS

Demographics is enabled by placing a statement as defined in Syntax at the beginning of:

- The submitted job
- A JDT or XJT
- A VI Compose configuration file (xgf.def)

The demographics command syntax is placed in the SETPARAMS array.

### SETPARAMS in target\_name files

The SETPARAMS command should be specified in only one of these files to ensure the creation of at most one of each of the target\_name files see definition below in either the first folder specified by SETMPATH or the local project folder. For FreeFlow Print Server, the target\_name file is written to the folder that begins with /opt/XRXnps in SETMPATH.

### When the SETPARAMS command precedes a SETPROJECT command

The target\_name file will be created in the first folder specified by SETMPATH. When the SETPARAMS command follows a SETPROJECT command, the target\_name file will be written to the local project folder. For FreeFlow Print Server, the target\_name file is always written to the folder that begins with / opt/XRXnps in SETMPATH. Xerox recommends placing SETPARAMS before SETPROJECT to ensure that all .vpd files reside in the same folder.

## Syntax

```
[ /Demographics [ option (target_name) ] ] SETPARAMS
```

```
[ /Demographics [ option [ (target_name) (foldername) (projectname) ] ] ] SETPARAMS
```

## Where:

- **option** is a five digit number in this format: IOCPD. Each digit has the following meaning:

**I** is the imaging request, use one of these values:

- **0** the job is not imaged during the demographics run
- **1** the job is imaged during the demographics run

**O** is the output format request, use one of these values:

- **0** none
- **1** Produce a database file (extension .vpr)
- **2** Produce an XML file (extension .vpd)
- **3** Produce both files

**C** is the contents request, combine one or more of these values:

- **0** none
- **+1** Resource Checking information
- **+2** Demographics information
- **+4** ACCLOG information refer to ACCLOG in the *VIPP® Language Reference Manual*

**P** is the print request. For a report at the end of the job combine one or more of these values (currently only the .vpr file can be printed. Therefore option "O" must be set to **1** or **3**):

- **0** none
- **+1** request printing of the demographics report (using reslist.dbm)
- **+2** request printing of the ACCLOG data report (using acclog.dbm)

**D** is the Demographics processing option, use one of these values:

- **0** disabled
- **1** enter demographics/resource checking mode and produce the output file(s) at the end of a line mode or database mode job
- **2** enter demographics/resource checking mode and produce the output file(s) only when XGFEND is executed

- **target\_name** is the base name of the file in which the demographics and resources information will be stored.
  - The extension .vpr will be automatically appended to it to create the database filename.
  - The extension .vpd will be automatically appended to it to create the XML filename.
  - If the job is run on VIeC in multi-instances mode a time stamp will also be appended to target\_name to ensure file name uniqueness if several jobs using the same target\_name are processed by VIeC in parallel.

By default, the files are created in one of these folders:

- The first writable folder specified by SETMPATH in xgfunix.run, or the project folder in project mode.
- For FreeFlow Print Server, the folder that begins with /opt/XRXnps in SETMPATH.

If the files already exist in one of the folders listed by SETMPATH or SETPPATH in project mode, they are overwritten. If all folders or existing files are read-only a VIPP\_access\_denied error will occur.

- **foldername/projectname** are additional parameters used to run demographics on a legacy job. The legacy job will be converted to a project identified by foldername and projectname using an appropriate tool.

**foldername** and **projectname** will be stored in the .vpd file, and are the names of the folder and project directories that will receive the resources when the job is converted to a project.

#### Example

This example enables demographics and resource checking information gathering and produces both a database (job1.vpr) and XML file (job1.vpd):

```
[ /Demographics [ 03301 (job1) ] ] SETPARAMS
```

## File samples

An example of a .vpr file and samples of ACCLOG and demographics data reports generated from the .vpr file are included here.

### TRAVEL\_US.VPR

```

TYPE,FLD1,FLD2,FLD3,FLD4,FLD5,FLD6,FLD7,FLD8,FLD9,FLD10,FLD11
H,23 Jun 05,projects,travel
2,DFE/RIP,Distiller
2,VII Version,VII_4.0
2,VSM Version,4.0
2,Option code,03731
2,RIP time,7
2,Caching,12
2,Resource Mode,Project
1,I0,NumberOfPages,100
1,I1,Media,
1,I2,DefaultMedia,100
1,I1,Plex,
1,I2,Simplex,100
1,I1,Faces,
1,I2,FacePrint,100
1,I1,PageSize,
1,I2,792_x_1224,100
1,I0,Copies,
1,I1,MinCopyCount,1
1,I1,MaxCopyCount,1
1,I0,NumberOfResources,41
1,I1,jdt,1
1,I1,dbm,1
1,I1,img,15
1,I1,mis,2
1,I1,fnt,22
0,Travel_US.jdt,jdt,0,1,0,1,1,1,0,D:\testcases\projects\travel\Travel_US.jdt,0
0,Travel_US.dbm,dbm,0,1,0,50,1,1,0,D:\testcases\projects\travel\Travel_US.dbm,0
0,covera,img,0,1,-1,15,1,2,1,D:\testcases\projects\travel\covera,0
0,coverb,img,0,1,-1,15,1,6,3,D:\testcases\projects\travel\coverb,0
0,coverc,img,0,1,-1,20,1,4,2,D:\testcases\projects\travel\coverc,0
0,Married,img,0,1,-1,14,1,4,2,D:\testcases\projects\travel\Married,0
0,NewsA,img,0,1,-1,15,1,1,1,D:\testcases\projects\travel\NewsA,0
0,NewsB,img,0,1,-1,15,1,5,3,D:\testcases\projects\travel\NewsB,0
0,NewsC,img,0,1,-1,20,1,3,2,D:\testcases\projects\travel\NewsC,0
0,Senior,img,0,1,-1,15,1,2,1,D:\testcases\projects\travel\Senior,0
0,Ship1,img,0,1,-1,17,1,2,1,D:\testcases\projects\travel\Ship1,0
0,Ship2,img,0,1,-1,17,1,4,2,D:\testcases\projects\travel\Ship2,0
0,Ship3,img,0,1,-1,16,1,6,3,D:\testcases\projects\travel\Ship3,0
0,Single,img,0,1,-1,21,1,18,9,D:\testcases\projects\travel\Single,0
0,TopA,img,0,1,-1,15,1,1,1,D:\testcases\projects\travel\TopA,0
0,TopB,img,0,1,-1,15,1,5,3,D:\testcases\projects\travel\TopB,0
0,TopC,img,0,1,-1,20,1,3,2,D:\testcases\projects\travel\TopC,0
0,1_Travel_US.ps,mis,0,1,-2,50,1,1,1,D:\testcases\projects\travel\1_Travel_US.ps,0
0,2_Travel_US.ps,mis,0,1,-2,50,1,2,1,D:\testcases\projects\travel\2_Travel_US.ps,0
0,Avalon,fnt,0,1,0,1,0,2,1,,0
0,Carmela,fnt,0,1,0,1,0,2,1,,0
0,Cochin,fnt,0,1,0,2,0,1,1,,0
0,Cochin-Bold,fnt,0,1,0,2,0,1,1,,0
0,Cochin-BoldItalic,fnt,0,1,0,2,0,1,1,,0
0,Cochin-Italic,fnt,0,1,0,1,0,1,1,,0
0,Courier,fnt,0,1,0,2,0,1,1,,0

```

```

0,Helvetica,fnt,6,53,0,2838,0,1,1,,0
0,Helvetica-Oblique,fnt,0,13,0,650,0,1,1,,0
0,NCR,fnt,0,4,0,4,0,1,0,,0
0,Poetica-ChanceryIV,fnt,0,1,0,2,0,1,1,,0
0,Poetica-SuppOrnaments,fnt,0,1,0,2,0,1,1,,0
0,Symbol,fnt,0,1,0,2,0,1,1,,0
4,0,1,RECORD
4,1,1,Customer
4,2,0,FullName,Pedro Martine
4,2,0,Destination,Alaska
4,1,1,Address
4,2,0,Street,100 Highland Ave
4,2,0,City,Boston
4,2,0,State,MA
4,0,1,RECORD
4,1,1,Customer
4,2,0,FullName,Marie Lyons
4,2,0,Destination,the Caribbean
4,1,1,Address
4,2,0,Street,543 Marsh Rd.
4,2,0,City,Boston
4,2,0,State,MA
4,0,1,RECORD
4,1,1,Customer
4,2,0,FullName,Rebecca Hall
4,2,0,Destination,the Bahamas
4,1,1,Address
4,2,0,Street,1234 Union St.
4,2,0,City,Boston
4,2,0,State,MA
4,0,1,RECORD
4,1,1,Customer
4,2,0,FullName,Kent Evans
4,2,0,Destination,Alaska
4,1,1,Address
4,2,0,Street,536 London Rd.
4,2,0,City,Boston
4,2,0,State,MA
4,0,1,RECORD
4,1,1,Customer
4,2,0,FullName,April Allen
4,2,0,Destination,the Caribbean
4,1,1,Address
4,2,0,Street,123 Monroe Ave.
4,2,0,City,Mobile
4,2,0,State,AL
4,0,1,RECORD
4,1,1,Customer
4,2,0,FullName,Anita Campbell
4,2,0,Destination,the Bahamas
4,1,1,Address
4,2,0,Street,400 East Ave.
4,2,0,City,Mobile
4,2,0,State,AL
4,0,1,RECORD
4,1,1,Customer
4,2,0,FullName,Pat Campbell
4,2,0,Destination,Alaska
4,1,1,Address
4,2,0,Street,100 Vic Park
4,2,0,City,Bangor
4,2,0,State,MI
4,0,1,RECORD
4,1,1,Customer
4,2,0,FullName,Linda Baker
4,2,0,Destination,the Caribbean
4,1,1,Address
4,2,0,Street,155 Fox Hollow
4,2,0,City,Bangor
4,2,0,State,MI
4,0,1,RECORD
4,1,1,Customer
4,2,0,FullName,Julie Allen
4,2,0,Destination,the Bahamas
4,1,1,Address
4,2,0,Street,123 Elm Street
4,2,0,City,Bangor
4,2,0,State,MI
%%EOF

```

## ACCLOG REPORT 1

This is a sample of an ACCLOG data report.

## FreeFlow VI Interpreter Accounting log Report on 23 Jun 05

### Project: projects/travel

Page 1

DFE/RIP ..... Distiller  
 VII Version ..... VII\_4.0  
 VSM Version ..... 4.0  
 Option code ..... 03731  
 RIP time ..... 7  
 Caching ..... VIPP full (12)  
 Resource Mode ..... Project

Label	Contents
<b>RECORD</b>	
Customer	
FullName	Pedro Martine
Destination	Alaska
Address	
Street	100 Highland Ave
City	Boston
State	MA
<b>RECORD</b>	
Customer	
FullName	Marie Lyons
Destination	the Caribbean
Address	
Street	543 Marsh Rd.
City	Boston
State	MA
<b>RECORD</b>	
Customer	
FullName	Rebecca Hall
Destination	the Bahamas
Address	
Street	1234 Union St.
City	Boston
State	MA
<b>RECORD</b>	
Customer	
FullName	Kent Evans
Destination	Alaska
Address	
Street	536 London Rd.
City	Boston
State	MA

## ACCLOG report 2

This is a sample of an ACCLOG data report.

**FreeFlow VI Interpreter Accounting log Report on 23 Jun 05**  
**Project: projects/travel**

Page 2

Label	Contents
<b>RECORD</b>	
Customer	
FullName	<b>April Allen</b>
Destination	<b>the Caribbean</b>
Address	
Street	<b>123 Monroe Ave.</b>
City	<b>Mobile</b>
State	<b>AL</b>
<b>RECORD</b>	
Customer	
FullName	<b>Anita Campbel</b>
Destination	<b>the Bahamas</b>
Address	
Street	<b>400 East Ave.</b>
City	<b>Mobile</b>
State	<b>AL</b>
<b>RECORD</b>	
Customer	
FullName	<b>Pat Campbel</b>
Destination	<b>Alaska</b>
Address	
Street	<b>100 Vic Park</b>
City	<b>Bangor</b>
State	<b>MI</b>
<b>RECORD</b>	
Customer	
FullName	<b>Linda Baker</b>
Destination	<b>the Caribbean</b>
Address	
Street	<b>155 Fox Hollow</b>
City	<b>Bangor</b>
State	<b>MI</b>
<b>RECORD</b>	
Customer	
FullName	<b>Julie Allen</b>
Destination	<b>the Bahamas</b>
Address	
Street	<b>123 Elm Street</b>
City	<b>Bangor</b>
State	<b>MI</b>
<b>End of Report</b>	

## DEMOGRAPHICS REPORT 1

This is a sample of a demographics data report.

FreeFlow VI Interpreter Demographics Report on 23 Jun 05 projects/travel

Page 1

Summary		Totals	
NumberOfPages .....	100	DFE/RIP .....	Distiller
Media		VII Version .....	VII_4.0
DefaultMedia .....	100	VSM Version .....	4.0
Plex		Option code .....	03731
Simplex .....	100	RIP time .....	7
Faces		Caching .....	VIPP full (12)
FacePrint .....	100	Resource Mode .....	Project
PageSize			
792_x_1224 .....	100		
Copies			
MinCopyCount .....	1		
MaxCopyCount .....	1		
NumberOfResources .....			
jdt .....	1		
dbm .....	1		
img .....	15		
mis .....	2		
fnt .....	22		

Resource Name	Type	Counts			Min/Max	First	First	Path	Status
		Ac.	Us.	Ca.	per Page	Page	Booklet		
Travel_US.jdt	jdt	1	1		0/1	1	0	D:\testcases\projects\travel\Travel_US.jdt	Ok
Travel_US.dbm	dbm	1	50		0/1	1	0	D:\testcases\projects\travel\Travel_US.dbm	Ok
covera	img	1	15	-1	0/1	2	1	D:\testcases\projects\travel\covera	Ok
coverb	img	1	15	-1	0/1	6	3	D:\testcases\projects\travel\coverb	Ok
coverc	img	1	20	-1	0/1	4	2	D:\testcases\projects\travel\coverc	Ok
Married	img	1	14	-1	0/1	4	2	D:\testcases\projects\travel\Married	Ok
NewsA	img	1	15	-1	0/1	1	1	D:\testcases\projects\travel\NewsA	Ok
NewsB	img	1	15	-1	0/1	5	3	D:\testcases\projects\travel\NewsB	Ok

## DEMOGRAPHICS REPORT 2

This is a sample of a demographics data report.

## FreeFlow VI Interpreter Demographics Report on 23 Jun 05 projects/travel

Page 2

Resource Name	Type	Counts			Min/Max per Page	First Page	First Booklet	Path	Status
		Ac.	Us.	Ca.					
NewsC	img	1	20	-1	0/1	3	2	D:\testcases\projects\travel\NewsC	Ok
Senior	img	1	15	-1	0/1	2	1	D:\testcases\projects\travel\Senior	Ok
Ship1	img	1	17	-1	0/1	2	1	D:\testcases\projects\travel\Ship1	Ok
Ship2	img	1	17	-1	0/1	4	2	D:\testcases\projects\travel\Ship2	Ok
Ship3	img	1	16	-1	0/1	6	3	D:\testcases\projects\travel\Ship3	Ok
Single	img	1	21	-1	0/1	18	9	D:\testcases\projects\travel\Single	Ok
TopA	img	1	15	-1	0/1	1	1	D:\testcases\projects\travel\TopA	Ok
TopB	img	1	15	-1	0/1	5	3	D:\testcases\projects\travel\TopB	Ok
TopC	img	1	20	-1	0/1	3	2	D:\testcases\projects\travel\TopC	Ok
1_Travel_US.ps	mis	1	50	-2	0/1	1	1	D:\testcases\projects\travel\1_Travel_US.ps	Ok
2_Travel_US.ps	mis	1	50	-2	0/1	2	1	D:\testcases\projects\travel\2_Travel_US.ps	Ok
Avalon	fnt		1		0/1	2	1		Ok
Carmela	fnt		1		0/1	2	1		Ok
Cochin	fnt		2		0/1	1	1		Ok
Cochin-Bold	fnt		2		0/1	1	1		Ok
Cochin-BoldItalic	fnt		2		0/1	1	1		Ok
Cochin-Italic	fnt		1		0/1	1	1		Ok
Courier	fnt		2		0/1	1	1		Ok
Helvetica	fnt		2838		6/53	1	1		Ok
Helvetica-Oblique	fnt		650		0/13	1	1		Ok
NCR	fnt		4		0/4	1	0		Ok
Poetica-ChanceryIV	fnt		2		0/1	1	1		Ok
Poetica-SuppOrnaments	fnt		2		0/1	1	1		Ok
Symbol	fnt		2		0/1	1	1		Ok

End of Report



