

VSI-FAX 4

INTEGRATION MANUAL

Copyright and Trademark Notices

Copyright © 1989 - 2000 by VSI, an Esker Company. All rights reserved.

VSI-FAX, DirectFax and CoverMaker are trademarks of VSI.

X and X/Open are trademarks of and UNIX is a registered trademark licensed exclusively by X/Open Company, Limited.

Microsoft, MS-DOS, Windows, Windows 95, Windows 98, Windows NT and Windows 2000 are registered trademarks of Microsoft Corporation.

Lotus, Lotus Notes and Domino are registered trademarks of Lotus Development Corporation.

Acrobat and Portable Document Format (PDF) are registered trademarks of Adobe Systems Incorporated (www.adobe.com).

WebHelp is a registered trademark of eHelp Corporation (www.ehelp.com).

VSI-FAX contains portions of Vermont Views software, copyright 1988 - 1990 Vermont Creative Software. All rights reserved. Vermont Views is a registered trademark of Vermont Creative Software.

VSI-FAX uses the Apache web server, Copyright (c) 1995-1999 The Apache Group. All rights reserved.

All other products or services mentioned in this document are trademarks or registered trademarks or service marks of their respective holders. VSI disclaims any responsibility for specifying which marks are owned by which holders.

The information contained in this document is subject to change without notice. Every effort has been made to provide accurate information. However, VSI does not guarantee the accuracy, adequacy or completeness of any information and is not responsible for any errors or omissions, or the results obtained from use of such information.

No part of this publication may be reproduced, stored in a retrieval system or transmitted in any form or by any means (mechanical, photocopying, recording, facsimile or otherwise) without prior written permission from VSI.

VSI, an Esker company

Voice: (800) 556-4874, (949) 462-2200

Fax: (949) 462-3300

Email: info@vsi.com

World Wide Web: <http://www.vsi.com>

TABLE OF CONTENTS

Foreword	5
Scope and Intended Audience	5
Typeface Conventions	5
A Word About Unix, Linux and Windows Notation	6
Notes	6
Obtaining Additional VSI-FAX Manuals	7
Additional Technical Information	7
Fundamentals	1
The “Fax Envelope”	2
Creating the Fax Envelope	3
Delivering the Fax Envelope	4
Integration Techniques	5
Which Integration Technique is Best for Me?	6
vfx Commands	7
autosend Directory	9
Email-to-Fax	11
Print-to-Fax	12
xmlf Commands	13
Software Development Kit (SDK)	14
WordPerfect Macros	15
lp Spooler	16
Using vfx Tags	17
The vfx Command Line	17
Tag Files	18
“Here” Documents	18

Embedded Commands	21
Embedded Images	21
Capabilities and Limitations	21
Image Positioning	22
Examples	22
Embedded Tags	23
Capabilities and Limitations	23
Examples	24
Print-to-Fax	25
Using VSI Print Drivers	25
Using Conventional Print Drivers	26
Email-to-Fax	27
Configuration and Setup	27
Message Formats	29
Fax Number In Subject Line	29
No Fax Number In Subject Line	30
Advanced Group Faxing	31
Which Group Faxing Strategy is Best For Me?	31
Fax Merge	32
Word Processor	32
Print Driver	34
Delivery Mechanisms	34
Fax Server	35
Batch Files	35
File Structure and Syntax	35
Delivery and Processing	36
Group Files	36
File Structure and Syntax	36
Delivery and Processing	37
Fax Merge Using Group and Template Files	37
Forms Overlay	39
Single Overlay	40
Multiple Overlays	41
Selective Merging	42
Creating Electronic Business Forms	43
Manual Fax Scanning	43
Scanner Support	44
Creating Faxable Signature and Logo Files	44

Virtual Fax Server	47
Capabilities and Limitations	47
Requirements	48
Ip Spooler	49
Creating a Printer Destination for Faxing	49
Creating a Custom Printer Interface Script	50
Sample Interface Script	51
WordPerfect Integrations	53
X Windows Versions	53
Files	53
Configuring the Integration	53
Using the Integration	54
Character-Based Versions	54
Files	54
Configuring the Integration	54
Using the Integration	55
Notes	55
Using XML-F	57
The xmlf Command	58
Fax-Submit Document	59
Fax-Submit-Response Document	62
Fax-Status Document	63
Fax-Status-Response Document	64
Fax-Cancel Document	65
Appendix A – vfx Command Reference	67
Appendix B – XML-F DTDs	73
Fax Submit DTD	73
Fax Submit Response DTD	76
Fax Status DTD	77
Fax Status Response DTD	78
Fax Cancel DTD	80
Fax Cancel Response DTD	81
Fax Resource DTD	82
Fax Resource Response DTD	83
Index	85

FOREWORD

Scope and Intended Audience

This manual provides detailed information about how to integrate VSI-FAX features and functions into existing business applications. We expect that persons integrating and fax-enabling applications already possess detailed technical knowledge about their existing hardware and software.

Refer to your *VSI-FAX Getting Started Manual* for product installation instructions. Also refer to your *VSI-FAX Server Manual* for detailed fax server maintenance and administration information.

Typeface Conventions

The following tables show examples of standard typeface styles used in this manual to convey various kinds of information.

EXAMPLE	DESCRIPTION
Click OK . - or - Choose File > Close .	Bold text is used to show actual Graphical User Interface (GUI) menus, commands and buttons. Also note that in the second example, sequential commands are separated by a greater-than (>) character.
Enter vfxadmin...	Bold monospaced text is used to show literal user input that must be entered exactly as it appears in the manual.
<my_filename> - or - <my_password>	Bold monospaced text inside angle brackets shows a type of user input (not a literal user input). Your actual entry will be your file or your personal password.

EXAMPLE	DESCRIPTION
Installation Complete. - or - ...set the VSIFAX variable... - or - ...are stored in \$VSIFAX/spool/dbs	Regular monospaced text is used for file, directory and environment variable names, as well as file entries and displayed messages.

A Word About Unix, Linux and Windows Notation

This manual supports Unix, Linux and Windows platforms. Whenever possible, meaningful examples are provided in all applicable formats. However, in the interest of clarity, directories and filenames are usually given in only one format. In most cases, these are interchangeable between platforms if you remember the following:

Unix and Linux environment variables are prefixed with a dollar sign (\$) ; Windows environment variables are enclosed in percent signs (%).

Unix and Linux path statements use forward slashes (/) ; Windows path statements use backlashes (\).

The following path statements are equivalent::

\$VSIFAX/MY_DIR/my_file	Unix and Linux
%VSIFAX%\MY_DIR\my_file	Windows

Notes

The following kinds of notes appear in this manual:

NOTE: This is a general note. We strongly suggest that you read these as they always contain important information you should be aware of before performing some action.

IMPORTANT: This is a warning. Warnings always contain information that if not heeded could result in unpredictable behavior or loss of data.

TIP: Tips present optional information intended to speed up your work or otherwise enhance your experience with our product. Tips never contain information that will cause a failure if ignored.

Obtaining Additional VSI-FAX Manuals

The VSI-FAX manuals are available in Adobe Acrobat (.pdf) format in the /docs directory of your installation CD. You can also download manuals in Acrobat format from the VSI FTP site as follows:

1. Connect to ftp.vsi.com.
2. Login in as anonymous
3. Go the /pub/releases/vsifax4 directory.
4. Go the directory for your version (e.g., /latest or /V400).
5. Go to the /docs directory.
6. Download the .pdf files.

You must have Adobe Acrobat Reader 4.0 installed on your system to view these .pdf files. The Acrobat Reader is available free of charge from the Adobe web site at www.adobe.com

You can also contact VSI (sales@vsi.com) if you would like to purchase additional printed manuals.

Additional Technical Information

Additional technical information is available from the Technical Support area of the VSI web site. Point your browser to www.vsi.com/support, then go to the Knowledge Base.

FUNDAMENTALS

This chapter introduces and describes the fundamental concepts of integrating VSI-FAX with another application.

TIP: Subsequent chapters in this manual assume that you are familiar with these fundamental concepts. If you are not absolutely certain that you understand them, you should read this material before proceeding with any VSI-FAX integration.

When most people consider integrating VSI-FAX with another application, they usually want to add one or more of the following capabilities to their existing business application:

- Send a Fax*** Use VSI-FAX to send faxes directly from another application.
- Receive a Fax*** Use VSI-FAX to receive faxes, then hand them to another application for processing.
- Administration*** Implement programmatic control and maintenance of the VSI-FAX server.

Sending a fax is by far the most common capability added via integration. For that reason, this manual will focus primarily on this aspect of integration.

The “Fax Envelope”

Sending a fax is very similar to sending a letter via the postal system. Consider that when you send a letter, you must have at least the following:

- Something to send
- Someone to send it to

Typically, a message is written or typed on a piece of paper and placed inside an envelope. Of course, you can include various other items such as brochures, price lists, bills, contracts, etc. In fact, some postal mailings can be rather large and contain many items in the same envelope. Conversely, if your message is very simple, you might just send a postcard. However, regardless of size or complexity, you still must have *something* to send.

After you have gathered all the items you want to send, placed them in the envelope, sealed it and addressed it to the recipient, you often have several different ways you can get it to the post office. For example, you can take it to your central post office in person or you can drop it in a local mail box and let a mail carrier to deliver it to the post office for you. You can even request delivery confirmation (usually by requiring that someone sign for the letter).

Similarly, when you send a fax, the minimum amount of information you need is:

- Something to fax
- A recipient (fax number or email address)

As with letters, faxes can contain other items (attachments) or be very simple (just a cover page). Once the various elements of the fax are defined, it is assembled into a “fax envelope” so that it can be sent. You also have several different ways to get it to the “post office” (i.e., fax server) and you can specify various kinds of confirmation (status).

Let’s set aside the analogies and metaphors for a moment and take a look at what fax envelopes really contain:

Content	Fax content is all the various things you want to fax to your recipient. It can be a simple text message or various files you want to image and send to this person.
Recipient Information	This is the person who will receive the fax. The minimum information required to send a fax is a fax number or an email address. However, you typically want to include other information such as the recipient’s name and company information in order to ensure that the right person gets the fax.
Cover Page (Sender Information)	Most faxes include a cover page. Cover pages are similar to return addressees on letters. They almost always include the sender’s information and usually include recipient’s addressing information. They can also include short messages. In fact, a fax can consist of only a cover page.
Send Parameters	Send parameters are various conditions you specify for sending this fax. For example, this is where you specify that the fax be sent from a specific fax device or sent at a specific time (e.g., delayed send).

Creating the Fax Envelope

There are two basic underlying methods for creating a fax envelope:

- **vfx** tags
- XML-F

It is important to understand that *all* VSI-FAX integration techniques build on one of these two basic methods.

vfx Tags

vfx tags are the most common way to define a fax envelope. These tags are nothing more than name/value pairs that are passed to the fax server.

Let’s consider a very simple fax envelope; a local fax number (555-1212) and a simple text file (`hello.txt`) containing the message “Hello, World.”

The **tfn** tag specifies the fax number. The actual tag for this fax envelope is:

tfn=555-1212

The **f11** tag specifies a file attachment. The actual tag for this fax envelope is:

f11=hello.txt

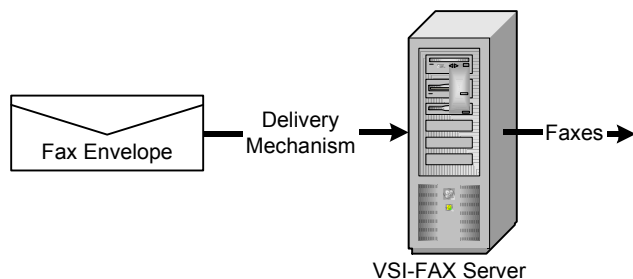
Currently, there are more than seventy different tags you can use to define various elements of a fax envelope.

XML-F

Extensible Markup Language (XML) is a language specifically designed to share data among various applications. Within the entire XML universe are various industry-specific specializations. One of these specializations is XML-F. The XML-F specification comprises eight public document types that can be used to define a fax envelope and communicate with a fax server.

Delivering the Fax Envelope

Once you have defined your fax envelope, you must deliver it to the fax server so that it can be imaged into a fax and sent via one of your fax devices.



Three things are needed to deliver a fax envelope to the fax server:

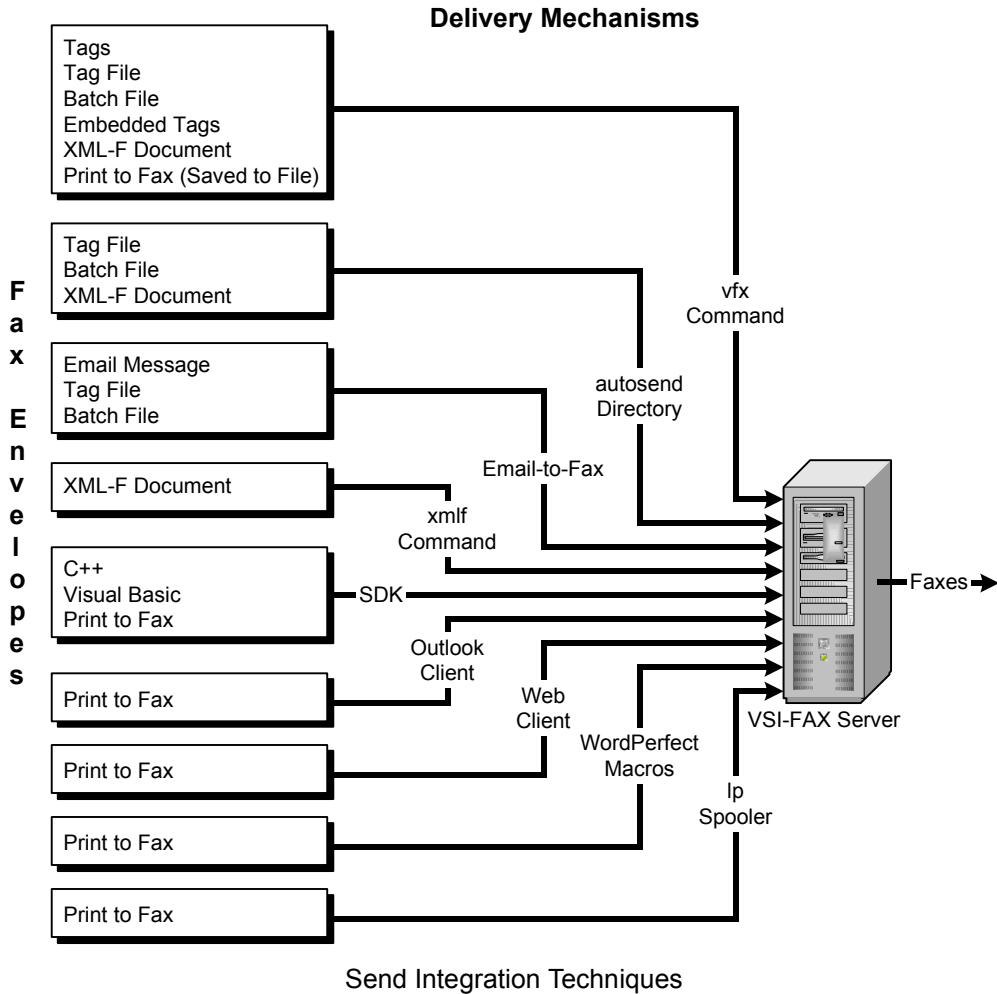
- Fax envelope
- Access to the fax server (host name, IP address, internet domain, email account)?
- VSI-FAX user account

VSI-FAX provides many different delivery mechanisms. In fact, virtually any integration situation will have access to at least one of these delivery mechanisms.

Integration Techniques

When you combine a specific way of creating a fax envelope with a specific delivery mechanism, you have a complete fax integration technique.

This figure shows all the various ways a fax envelope can be created and the various delivery mechanisms available for each:



Another way to look at integration solutions is to consider a matrix that lists ways to create fax envelopes as rows and delivery mechanisms are columns.

		DELIVERY MECHANISMS								
		vfx Command	autosend Directory	Email-to-Fax	xmlf Command	Fax SDK	Outlook Fax Client	Web Fax Client	WordPerfect Macros	Ip Spooler
WAYS TO CREATE FAX ENVELOPES	Tags	X								
	Tag Files	X	X	X						
	Batch Files	X	X	X						
	Embedded Tags	X								
	XML-F Document	X	X		X					
	Email Message			X						
	C++					X				
	Visual Basic					X				
	Print to Fax	X				X	X	X	X	X

Each way of creating a fax envelope and each delivery mechanism will be discussed in greater detail elsewhere in the manual.

Which Integration Technique is Best for Me?

As you can see from previous topics, there are many different ways to integrate VSI-FAX into your existing business application. Some integrations solutions are better suited to a distributed environment, while others are relatively simple and easy to implement, still others require prior programming experience.

In order to help you choose the best integration strategy for your needs, you should consider the following:

- Is your application client/server (does it run on a different network node than the fax server)?
- How automated do you need your integration to be?
- Do you require fax status or notification?
- What are your programming skills?
- Is purchasing additional VSI-FAX licenses an issue?

vfx Commands

vfx commands are a simple and powerful delivery mechanism. **vfx** is simply a command line utility that provides many options for precisely defining a fax envelope. One of these options is a tag statement (**-t <tag>=<value>**). There are currently more than seventy different tags that can be specified in addition to other **vfx** options.

Advantages

Very powerful and flexible. Provides maximum options for creating the fax envelope and full support for all content parameters, send options, fax status and notifications.

Excellent choice if your application can execute external commands and is running locally on the fax server.

It is still an acceptable choice if your application runs remotely (i.e., from another IP address than the fax server). However, you must purchase an additional Virtual Fax Server license and install the Virtual Fax Server on the remote computer or workstation.

Creates a fax request ID you can use to track fax status.

Disadvantages

Additional cost involved with remote integrations - you must purchase an additional Virtual Fax Server license and have a network connection to the fax server.

Poor choice if your application cannot execute external commands.

Ways to Create Fax Envelopes

- | | |
|--------------------|--|
| Tags | Use the vfx -t option to supply a series of tag statements (<tag>=<value>) on the vfx command line. This normally requires that some sort of macro or script be run from inside your application. |
| Tag Files | Create an ASCII text file containing various tag statements (<tag>=<value>), then pass this "tag file" to the fax server using the vfx -c <tag_file> command. Tag files are processed just as if the tag statements were entered directly in a command shell. |
| Batch Files | Similar to tag files, but only appropriate when you want to send multiple personalized faxes to multiple recipients.

Batch files are divided into sections. A common section contains information sent to all recipients; various recipient sections contain personalized information for that fax recipient. |

Embedded Tags	<p>Most often used in print-to-fax situations. The fax server will automatically scan PCL and PostScript files for embedded tags. If it finds them, it will process them just as if they were enter in a command shell.</p> <p>A common way to use embedded tags is to enter embedded tag statements directly into a word processor document, print the document to a PCL or PostScript file, then pass the file to the fax server as a vfx command line argument.</p>
XML-F Documents	<p>XML-F provides eight different Document Type Descriptions (DTDs) that describe how to create various files you can pass to the fax server and the responses you can expect back from the fax server.</p> <p>XML is a completely open data exchange standard. Furthermore, it is completely platform and delivery mechanism neutral. This makes it an excellent choice for heterogeneous environments and large enterprise-wide faxing solutions.</p>
Print-to-Fax (Saved to File)	<p>A simple approach that does not lend itself to full automation. However, it works with almost any application. Open the application, print to a PCL or PostScript file, then pass the file to the fax server as a vfx command line argument.</p> <p>However, if you do not specify recipient information with embedded tags, you must supply recipient tags on the vfx command line.</p>

See Also

Refer to *Appendix A – vfx Command Reference* (page 67) for complete documentation for the **vfx** command.

If you will be creating your fax envelopes via tags, tag files, batch files, embedded tags or print-to-fax:

- *Using vfx Tags* (page 17)

If you will be creating your fax envelopes via XML-F:

- *Using XML-F* (page 57)
- *Appendix B – XML-F DTDs* (page 73)

If the application you are integrating does not run on the fax server network node:

- *Virtual Fax Server* (page 47)

autosend Directory

The autosend directory is a specific file system directory that is continuously monitored by the fax server. When the fax server finds a tag file, batch file or XML-F document, it uses it to send a fax just as if the file had been passed to it via a **vf**x command (page 7).

The location of the autosend directory is `$VSI/FAX/autosend`.

Files placed in the autosend directory must have the proper file extension:

- Tag files must have a `.tag` extension
- Batch files must have a `.bat` extension
- XML-F files must have a `.xml` extension

Advantages

Excellent choice for situations where you do not have a network or email connection to the fax server but do have access to the fax server file system (e.g., via NFS).

Disadvantages

You are limited to creating fax envelopes with tag files, batch files or XML-F.

This is a “blind send.” Therefore:

- Fax status and notifications are not available
- If using XML-F, only the Fax Submit DTD is applicable
- If errors occur, your fax won’t be sent and you will not receive any notification whatsoever that this occurred

Ways to Create Fax Envelopes

Tag Files	Create an ASCII text file containing various tag statements, then copy or move this “tag file” to the fax server autosend directory (<code>\$VSI/FAX/autosend</code>).
Batch Files	<p>Similar to tag files, but only appropriate when you want to send multiple personalized faxes to multiple recipients.</p> <p>Batch files are divided into sections. A common section contains information sent to all recipients; various recipient sections contain personalized information for that fax recipient.</p>
XML-F Documents	Create a file according to the XML-F Fax Submit DTD (page 73), then move this XML-F document to the fax server autosend directory (<code>\$VSI/FAX/autosend</code>).

See Also

If you will be creating your fax envelopes via tags, tag files, batch files, embedded tags or print-to-fax:

- *Using vfx Tags* (page 17)

If you will be creating your fax envelopes via XML-F:

- *Using XML-F* (page 57)
- *Appendix B – XML-F DTDs* (page 73)

Email-to-Fax

When a fax server is networked to a mail server, you can configure a dedicated email account (typically “autofax”) for the express purpose of sending faxes. Then you can configure the fax server to periodically poll this email account. When the fax server receives email messages at the “autofax” account, it assumes that they are faxes waiting to be sent and processes the email messages accordingly.

Advantages

Excellent choice for situations where you do not have a network connection to the fax server or have access to the fax server file system but do have an email connection to the fax server via a POP3 mail gateway.

Disadvantages

Your application must be able to send an email message.

You must have an incoming (POP3) internet mail server and it must be networked to the fax server.

Ways to Create Fax Envelopes

Email Message

If you include a valid fax number or email address on the subject line, the fax server assumes that the remainder of the email message (including any file attachments) is the fax content.

The syntax for including a fax number in the email subject line is:

<fax_number> [<subject_text>]

Where **<fax_number>** is a valid fax number or internet email address and **<subject_text>** is optional subject line text.

Tag and Batch Files

If you do not include a valid fax number or email address on the subject line, the fax server assumes that the remainder of the email message is a tag or batch file.

See Also

Refer to *Email-to-Fax* (page 27) for a complete discussion of how to setup, configure and use the email-to-fax delivery mechanism

If you will be including tag or batch files in the body of your email message:

- *Using vfx Tags* (page 17)

Print-to-Fax

Print-to-fax is an extremely versatile way to either deliver a fax envelope to the fax server or prepare a fax envelope so that it can be delivered by another means. While this integration technique is hard to fully automate, it can be used to fax-enable virtually any application with printing capabilities.

Advantages

Can be used to fax-enable virtually any application on any platform as long as it can print to file.

If one of the VSI print drivers is used, users will be prompted to enter send information (e.g., fax number, recipient name) and the fax envelope will be delivered to the fax server.

You can direct your application's output to any PostScript, PCL or ASCII print driver - you are not restricted to using one of the VSI print drivers.

Disadvantages

Difficult to fully automate.

If you do not use one of the VSI print drivers, you must get the PostScript, PCL or ASCII file to the fax server - you do not have an integrated delivery mechanism. In most cases, you will need to pass these files to the fax server using **vfx** commands (page 17).

Ways to Create Fax Envelopes

Tags	If you do not specify send information via embedded commands, you must do so when you deliver the output file to the fax server. In these cases, you must at least supply a fax number using the fnn tag.
Template Files	You can also use template files in place of tag files or embedded commands to create the fax envelope. This is especially useful for group faxing.

See Also

- *Print-to-Fax* (page 25)
- *Using vfx Tags* (page 17)

xmlf Commands

Extensible Markup Language (XML) is emerging as the preferred technique for sharing data among applications. This is especially true for applications that collect and transport data via the internet.

Advantages

Any XML-aware application can easily be configured to output and receive XML documents.

XML is inherently platform and transport independent. This is especially useful for integrating existing business applications running on platforms not supported by VSI-FAX (e.g., VMS).

Disadvantages

Fax envelopes must be created using the XML-F Fax Submit documents (page 73).

If your application cannot issue external commands, you must provide scripting to issue **xmlf** commands or place the send document in the autosend directory (page 9).

Ways to Create Fax Envelopes

XML-F documents are the only way to create fax envelopes for use with the **xmlf** delivery mechanism. The application sends a fax by writing an XML Fax Submit documents (page 73) and passing it to the fax server via the **xmlf** command (page 58).

See Also

- *Using XML-F* (page 57)
- *Appendix B – XML-F DTDs* (page 73)

Software Development Kit (SDK)

The VSI Software Development Kit (SDK) provides various Component Object Model (COM) classes and sample code that can be used by software developers to integrate fax functionality into Windows applications. These applications can be written using Microsoft Visual Basic 6.0 or Microsoft Visual C++ 6.0 and will run on Windows 95, Windows 98, Windows NT 4.0 or Windows 2000 platforms.

Advantages

Experienced C++ and Visual Basic programmers can easily leverage existing knowledge to create powerful fax integrations.

Disadvantages

Only works on supported Windows platforms. Unix/Linux users generally integrate via **vfxf** (page 7) or **xm1f** (page 13) commands rather than at the API level.

Ways to Create Fax Envelopes

- | | |
|---------------------|---|
| C++ | Add the fax automation classes to your C++ projects and use them to create various custom fax controls and Graphical User Interfaces (GUIs). |
| Visual Basic | Add fax automation classes to your the Visual Basic projects and use them to create various custom fax controls and GUIs. |
| Print-to-Fax | <p>After you have successfully installed the VSI-FAX print driver using the InstallPrinter program, you can send faxes from within any Windows application that supports normal printing (virtually any Windows application).</p> <p>However, it is important to understand that VSI-FAX print driver only images the information a user wants to fax (via the File > Print... command). In order to send a fax, destination information still must be supplied. A sample visual basic application is provided that receives the imaged file from the print driver and displays a simple fax form for entering the fax destination information.</p> |

See Also

- *VSI SDK Online Help*

WordPerfect Macros

The VSI WordPerfect macros provide an easy way to fax enable various versions of WordPerfect running on Unix and Linux platforms. Separate macros and support libraries are provided for character-based and graphical versions of WordPerfect.

Advantages

Very easy to implement (complete out-of-the-box turnkey solution).

Disadvantages

These macros only work with WordPerfect for Unix/Linux. Windows users should use print-to-fax (page 12) with the Outlook or web fax client print drivers.

Hard to fully automate.

Ways to Create Fax Envelopes

The VSI WordPerfect macros image pages in the current WordPerfect session, prompt the user for fax destination information, then pass the fax envelope to the fax server via **vf~~x~~** commands (page 7). This is the only way to create a fax envelope for use with this delivery mechanism.

See Also

- *WordPerfect Integrations* (page 53)

lp Spooler

This is essentially another kind of print-to-fax integration (page 12) that uses a custom printer interface script instead of a fax print driver.

Advantages

This technique works on most (but not all) Unix and Linux platforms.

Disadvantages

This technique only works on certain Unix and Linux platforms. You cannot use it on Windows platforms.

Ways to Create Fax Envelopes

The most common method of lp faxing is to use the Unix **lp** command. Typically, the **lp** command accepts several command line options that work well for sending a fax. In order for lp faxing to work correctly, two arguments must be passed to the interface script:

- Fax number
- Name of the file you want to fax

See Also

- *lp Spooler* (page 49)

USING VFX TAGS

This chapter discusses various ways you can use **vfx** tags to fax-enable your existing business applications.

Tags are always three-character mnemonics for the function they accomplish. For example, the **tfn** tag is used to define the fax number, the **tgf** tag is used to define a tag file, etc.

Refer to your *VSI-FAX Server Manual* for a complete list of all available **vfx** tags.

The vfx Command Line

The easiest way to use tags is to include them on with a **vfx** command by using the **-t** option. For example, to fax a text file (`hello.txt`) containing the message “Hello, World” to local fax number 555-1212, you would enter in a command shell at the fax server:

```
vfx -t tfn=555-1212 -t fl1=hello.txt
```

Passing tags to the fax server using **vfx** commands is a good integration technique for situations where your existing business application can issue external commands.

NOTE: The VSI-FAX WordPerfect for Unix macros use this integration technique. When these macros are run from inside a WordPerfect session, they prompt the user for the fax number, then issue an external **vfx** command that images the current document and passes it to the fax server.

Refer to *Appendix A – vfx Command Reference* (page 67) for complete **vfx** command documentation.

Tag Files

Tag files are ASCII text files that contain one **vfx** tag per line. For example, you could fax the same message as the previous example by saving this information to tag file `hello.tag`:

```
tfn=555-1212  
fll=hello.txt
```

To send the fax, you would pass the tag file to the fax server using the **-t tgf=<tag_file>** option as follows:

```
vfx -t tgf=hello.tag
```

Tag files are especially useful for situations where your existing business application can write output to a file but cannot issue external commands. However, you must write the script that passes this file to the fax server. This is typically done using **vfx** commands, placing the file in the `autosend` directory or emailing the file to the fax server.

IMPORTANT: Tag files have one important limitation: they cannot send different faxes to different recipients. You can send the same fax to more than one recipient (i.e., send a group fax) but the faxes will be identical. If you need to send different faxes to different recipients, consider using a batch file. Refer to *Advanced Group Faxing* (page 31) for additional information.

“Here” Documents

“Here” documents are special cases within tag files and batch files (page 35) where a local file, instead of being referenced externally, is actually embedded directly into the tag or batch file.

“Here” documents can be embedded using any of these **vfx** tags:

TAG	DESCRIPTION
fll	Local file.
ntf	Note file (memo).

The normal way to use any of these tags is to supply an external file specification (full path and file name you want to include). When you define a “here” document, you simply define an internal area of the tag or batch that contains the same information. Consider this example:

```
fll=<<EOF  
Hello, World.  
EOF
```

Instead of the **f11** tag referencing an external text file (containing the message, “Hello, World.”), the message is directly embedded using the **<<EOF** and **EOF** statements to define the area between them as a “here” document.

TIP: You are not limited to using End of File (EOF) character strings to define your “here” documents. You can use any character sequence you like, provided that it does not contain spaces and is not used in the body of your “here” document. In fact, you *will* need to use a different character sequence if your “here” document contains an EOF statement within it.

“Here” documents provide additional flexibility when you need to fax-enable an application that can write output to a file but cannot issue external commands.

EMBEDDED COMMANDS

You can embed **vf**x commands in ASCII text, PCL or PostScript files, then pipe or redirect them to the **vf**x command as standard input (STDIN). These “embedded commands” are processed just as if you had entered them directly on the command line.

Embedded commands are typically used to do these things:

- Embed images, such as company logos, directly into a file you want to fax
- Embed tags that contain fax envelope information (page 2) directly into a file you want to fax

Embedded commands are especially useful when integrators are programmatically generating a faxable file directly from an existing business application. These embedded commands allow you to create the entire fax (including the fax envelope information) and simply hand it to the fax server for sending - no additional formatting or intervention is required.

Embedded Images

The **@+IMAGE** embedded image command allows you to include TIFF files, such as your company logo or a personal signature, in a faxable document. Furthermore, you can control the exact position of each embedded image on the faxed document.

The basic syntax for an embedded image statement is:

```
@+IMAGE [<file_name.tif>]
```

Where **<file_name.tif>** is the name of your TIFF file.

Capabilities and Limitations

The “**@+IMAGE**” embedded command must be all uppercase. However, the remainder of the embedded image statement (e.g., file name, position coordinates, etc.) is case-insensitive.

The **@+IMAGE** statement can only include single-page TIFF image files. Multi-page TIFF files are not supported (only the first page will be faxed).

Embedded image statements can only refer to files in these directories:

- `$VSIFAX/lib/images`
- `$VSIFAX/faxq/<user_ID>/images`

Because these files must be located in one of these directories, full path names are not required nor are they supported.

Image Positioning

The default location for embedded images is the exact location of the embedded image statement in the text, PCL or PostScript file. In other words, if you placed your embedded image in the middle of a text, PCL or PostScript file, the top-left corner of that embedded image would appear in the middle of the faxed document. Subsequent text, PCL or PostScript data would appear after the embedded image file.

While this may be sufficient for some applications, in most cases more precise positioning is required. This is accomplished by including position coordinates in your embedded image statement as follows:

```
@+IMAGE [<filename.tif>;ax=<pos>{i | m};ay=<pos>{i | m}]
```

Where `ax=<pos>{i | m}` is the horizontal coordinate and `ay=<pos>{i | m}` is the vertical coordinate. Specify `i` if your coordinates are in inches; specify `m` if your coordinates are in millimeters.

Position coordinates define a horizontal and vertical offset from the upper left corner of the fax document, not from the default image location (i.e., the location of the embedded image statement in the text, PCL or PostScript file). In other words, if position coordinates are supplied, the position of the embedded image statement in the text, PCL or PostScript file is irrelevant.

Examples

This example shows how to embed a TIFF file (`logo.tif`) directly into a text, PCL or PostScript in the default location (i.e., at the exact point of the embedded image statement in the text, PCL or PostScript file). It will extend down from this point to the full height of the image and up to the full horizontal width of the page.

```
@+IMAGE [logo.tif]
```

This example shows how to embed a TIFF file `logo.tif` two inches from the left margin and five inches from the top margin. The image will extend down from this point to its full height and up to the full horizontal width of the page.

```
@+IMAGE [logo.tif;ax=2i;ay=5i]
```


Embedded Tags

Using vfx Tags (page 17) discussed various ways to use tags directly or to pass various text files containing tag statements to the fax server. There is yet another way to use **vfx** tags that does not require ASCII text files.

You can embed **vfx** tags directly into ASCII text, PCL or PostScript files. If **vfx** encounters specific embedded tag sequences in these files, it uses them to create the fax envelope.

Embedded tags are a good way to extend print-to-fax functionality (page 25) to implement more powerful fax integrations.

Fax Merge Fax merge (page 32) is a practical example of embedded tags. Basically, you begin by creating a source document in your word processor that populates various embedded tag statements via the word processor's "mail merge" feature. Next, you print this document to a PCL or Postscript file and pass it to the fax server. The fax server recognizes the embedded tags in the printer file and creates the fax envelopes accordingly.

The syntax for embedded tags is:

```
@+VFX [<tag>=<value>;<tag>=<value>;...]
```

Where **<tag>** is a **vfx** tag and **<value>** is the value assigned to that tag. Multiple tag/value pairs are separated by semi-colons.

You can include more than one embedded tag sequence in a PCL or Postscript file and each embedded tag sequence can include multiple **vfx** tags.

Capabilities and Limitations

Embedded tags are scanned only if the file is piped or redirected to **vfx** as STDIN.

The "@+VFX" statement must be all uppercase. However, the tag/value statements are case-insensitive.

The entire embedded tag sequence must be continuous - it cannot contain any spaces or explicit line breaks (hard returns).

If you need to include a space in a tag value, use the tilde (~) character. Any tilde (~) character encountered is replaced by a space unless it is escaped (\~).

You can only embed **vfx** tags that store fax envelope (page 2) information. These are sometimes referred to as "cover page" tags because their primary purpose is to place fax envelope information on fax cover pages. This is a list of tags you can embed:

- **cli** (client ID)
- **fa1** - **fa3** (from address lines 1 thru 3)
- **fcn** (from country name)
- **fco** (from company name)

- **fem** (from email address)
- **ffn** (from fax number)
- **fnm** (from name)
- **fvn** (from voice number)
- **ntf** (note file)
- **sub** (subject line)
- **sig** (signature file)
- **tco** (to company name)
- **tfn** (to fax number)
- **tg1** - **tg4** (user-defined tags 1 thru 4)
- **tin** (custom to information)
- **tnm** (to name)
- **tvn** (to voice number)

Refer to your *VSI-FAX Server Manual* for detailed information about these and other **vfx** tags.

Examples

This is an example of an embedded local fax number:

```
@+VFX[tfn=555-1212]
```

This is an example of an embedded local fax number, recipient name (with tilde (~) as a space holder) and cover page:

```
@+VFX[tfn=555-1212;tnm=Mr.~Smith;cvr=classic]
```

This example shows how to use both embedded tags and embedded images in a file (`sample.txt`):

```
@+VFX[tfn=5551212;tco=TechSupport]This is an example of how to send  
a fax using the embedded commands. The embedded command sends the  
fax number and the "To Company Name" to the VFX command.
```

```
@+IMAGE[<filename.tif>] This command places an image file in the  
location were the command is found.
```

You could redirect this file to the **vfx** command using any of the following commands:

```
vfx < sample.txt  
more sample.txt | vfx  
$ cat sample.txt | vfx  
C:\> type sample.txt | vfx
```

Note that the first two commands work on any platform, the third command only works on Unix/Linux and the fourth command only works on Windows NT.

PRINT-TO-FAX

Print-to-fax is an extremely versatile way to either deliver a fax envelope to the fax server or prepare a fax envelope so that it can be delivered by another means. While this integration technique is hard to fully automate, it can be used to fax-enable virtually any application with printing capabilities.

There are really two primary ways to fax-enable an application using print-to-fax:

- Use one of the VSI print drivers
- Use a normal (non-VSI) PostScript, PCL or ASCII print driver and save the output to a file

Using VSI Print Drivers

The easiest way to implement print-to-fax is to direct your application's print output to one of the fax print drivers provided with VSI-FAX. Typically, a separate print driver is provided with each fax client.

Once a fax client is successfully installed on a client computer or workstation, you can print to fax directly from any application running on that client computer or workstation simply by choosing the fax print driver is the print destination.

When you do this, the VSI print driver rasterizes your application information session, prompts you for essential send information (e.g., fax number, recipient name) and delivers the fax envelope to the fax server. In other words, the VSI print drivers provide both a way to create the fax envelope and a delivery mechanism for getting the envelope to the fax server.

Using Conventional Print Drivers

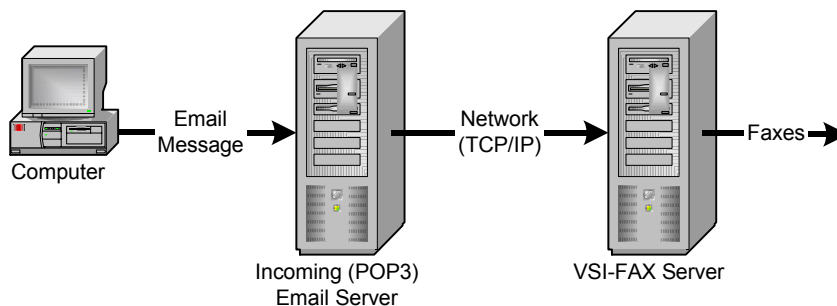
The second way to implement print-to-fax is to direct your application's print to a conventional (non-VSI) PostScript, PCL or ASCII print driver and save the output to a file.

The advantage to this approach is that you can use any PostScript, PCL or ASCII print driver installed on your system - you are not restricted to using one of the VSI print drivers.

The disadvantage to this approach is that you must get the PostScript, PCL or ASCII to the fax server - you do not have an integrated delivery mechanism. In most cases, you will need to pass these files to the fax server using **vfxx** commands (page 17).

EMAIL-TO-FAX

Email-to-fax is a simple and powerful delivery mechanism that is ideally suited to remote fax integration because you do not need a direct network connection to the fax server, nor do you need access to the fax server file system. All you need is a network connection from the fax server to an incoming (POP3) internet mail server. Of course, your application must be able to send an email message.



Configuration and Setup

The email to fax feature is normally setup during VSI-FAX installation. However, if for some reason you did not do this when your fax server was installed, you can do it now by performing the following:

1. Verify the integrity of the network connection between the fax server and your incoming (POP3) internet mail server.
2. Create a dedicated fax email account on your incoming (POP3) internet mail server.

TIP: We strongly recommend creating this account with the user name “autofax” as it is extremely unlikely that this name is already being used.

The remainder of this chapter will assume that you are using “autofax” for your POP3 email-to-fax account. If you decide to use another name, substitute that name in our examples.

IMPORTANT: This email account must be set aside strictly for faxing. The reason for this is that the fax server will assume that any message in this email inbox is a fax and will process it accordingly. Therefore, ensure that the “autofax” email address is not added to any email groups or corporate mailing lists.

3. Configure the fax server to monitor the “autofax” email account.

You can do this using the VSI MMC Fax Server Administration snap-in or by manually modifying settings in `$VSI\FAX\lib\vsisrv.ini` configuration file.

- (a) Enter your incoming (POP3) internet mail server host name, fully qualified internet domain name or IP address.

IF YOU ARE	DO THIS
Using the MMC snap-in	Go to the Fax Server Email Gateway properties and enter the host name, domain name or IP address in the Pop Host name field.
Modifying <code>vsisrv.ini</code>	Go to the VPOPD section and add the host name, domain name or IP address of your incoming (POP3) internet mail server to the <code>host-name=</code> entry.

- (b) Enter the name of the POP3 user account you created in step 1.

IF YOU ARE	DO THIS
Using the MMC snap-in	Go to the Fax Server Email Gateway properties and enter the user name in the User name field.
Modifying <code>vsisrv.ini</code>	Go to the VPOPD section and add the user name to the <code>user-name=</code> entry.

- (c) Enter the password for the POP3 user account (optional).

If you did not create a password for this account in step 1, you do not need to enter a password.

IF YOU ARE	DO THIS
Using the MMC snap-in	Go to the Fax Server Email Gateway properties and enter the password in the Password field.
Modifying <code>vsisrv.ini</code>	<p>IMPORTANT: The <code>vsisrv.ini</code> password entry is encrypted, therefore it cannot be directly added via a text editor. You must use the vfxadmin command line utility to add this entry.</p> <p>Open a command shell and enter the following:</p> <pre>vfxadmin config VPOPD password <entry></pre> <p>Where <entry> is the actual password defined for the POP3 user account you created in step 1.</p>

4. Complete

Message Formats

There are two different acceptable message formats for email messages sent to the “autofax” account.

Fax Number In Subject Line

If your email message to the “autofax” account includes a valid fax number or email address in the subject line, the fax server assumes that the body of the email message is plain text.

The syntax for including a fax number in the email subject line is:

<fax_number>[<subject_text>]

Where **<fax_number>** is a valid fax number or internet email address and **<subject_text>** is optional subject line text.

The remainder of the mail message is limited to ASCII text and other file attachments that can be natively imaged by the fax server (e.g., PostScript, PCL, text, etc.). For example, you cannot include tag or batch files in this kind of email message. The fax server will simply image the tag and batch files as plain text.

No Fax Number In Subject Line

If your email message to the “autofax” account does not include a valid fax number or email address in the email subject line, the fax server assumes that the body of the email message is a tag or batch file.

Refer to *Tag Files* (page 18) and *Batch Files* (page 35) for additional information about how to define a fax envelope using these techniques.

ADVANCED GROUP FAXING

Virtually any user can easily send the same fax to a group of recipients using one of our fax clients. It is even practical to do modest amounts of group faxing directly from the **vf**x command line.

However, if you need to send faxes to large groups, send personalized faxes tailored to each recipient or re-use your group faxes, you may want to consider one or more of the advanced techniques described in this chapter.

Which Group Faxing Strategy is Best For Me?

Before you decide on a particular approach or “strategy” for your group faxing, consider the following:

Are You Using Recipient Information From A Database Or Spreadsheet Program?

Consider using a group file (page 36). Simply export your information to a delimited ASCII text file. This strategy is ideal for semi-automated “fax blasting” to a large recipient list.

Are You Creating Your Fax Document With A Word Processor?

Consider using your word processor’s mail merge feature to create a fax merge (page 32). While you can’t fully automate this technique, most modern word processors have some sort of mail merge feature that can be used to send group faxes.

Do You Need to Re-Use Your Group Fax?

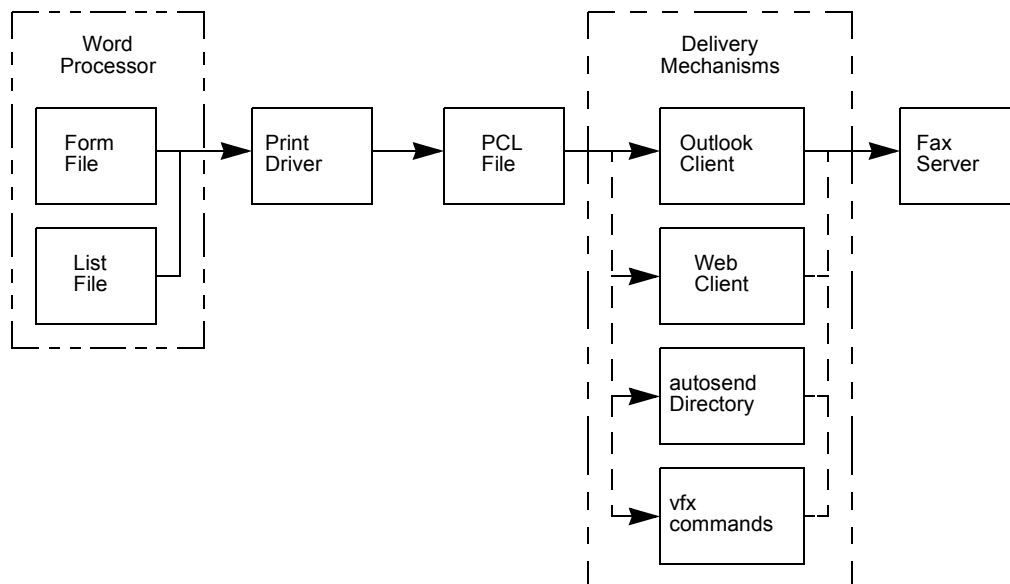
Consider using a template file. You can define the overall layout of your fax with the template file and populate it with recipient information from a group file (page 36). If you need to send another group fax to a different group of recipients, simply use a different group file.

Fax Merge

Fax merge is a special kind of “print to fax” operation (page 25) that uses your word processor’s “mail merge” feature to send individualized faxes to a group of recipients.

When you print a mail merge job, your word processor creates each individualized document by inserting variable information, such as the person's name, title and address, from a data source document (list file) into a content template document (form file). The software repeats this process as many times as necessary to create and print all the required individualized documents.

This diagram shows how to adapt a basic mail merge feature for fax merge.



Fax Merge Functional Block Diagram

Word Processor

The first step in creating a fax merge job is to use your word processor’s mail merge feature to create a content template (form file) that will be individualized with information from a data source (list file). In many cases, this data source is a simple text file containing entries for each fax recipient. However, some word processors support reading this recipient information directly from a database.

In order to adapt your word processor's mail merge feature for fax merging, you must create your form file using a very specific structure. This is because when you send a fax directly from your word processor, the fax server must be able to recognize certain information as “destination data” and the rest of the information as the actual fax you want to send.

Destination data is embedded into the form file using standard VSI embedded tags. Embedded fax tags always use this format:

```
@+TAG [<tag>=<value>]
```

Where <tag> is a valid send tag and <value> is a valid value for that tag. For example, this embedded tag is used to include a “classic” cover page with a fax:

```
@+TAG [cvr=classic]
```

Refer to *Embedded Tags* (page 23) for additional information.

IMPORTANT: All embedded tag statements must have the “Courier” font applied to them. Otherwise, the fax server will image the embedded tag statements as plain text.

Example Form and List Files

These are example Microsoft Word form and list files. They will be used to send a simple fax survey to Mr. Smith and Mr. Jones.

```
@+TAG [tfn=«FaxNumber»]  
@+TAG [cvr=classic]  
  
- - - Page Break - - -  
  
Hello Mr. «Name»,  
Do you still live in «Location»?
```

Form File

```
Name, Location, FaxNumber  
Smith, London, 5551212  
Jones, New York, 5551213
```

List File

Notice that the example form file contains an embedded fax number (@+TAG [tfn=«FaxNumber»]) and an embedded cover page designation (@+TAG [cvr=classic]), followed by a page break. This first page break tells the fax server that all the information in the document before the page break is destination data, not part of the actual fax. Also notice that the embedded fax number value is actually a field that will be read from the list file.

IMPORTANT: The entire embedded tag area of the form file (i.e., everything before the first page break) must have the “Courier” font applied to it. Otherwise, the fax server will image the embedded tag statements as plain text.

The remainder of the form file is the actual fax template that will be individualized for each recipient. This includes a salutation using a name field («Name») and the survey question using a location field («Location»).

The sample list file is a simple text file. Each data record is separated by a hard return; each value within that record is separated by a comma. Typically, the first record in a list file contains the header labels for the remaining records. In this case, header labels `Name`, `Location` and `FaxNumber` tell us that each subsequent record (line) will contain these three pieces of information (values) in this order.

This is how each person's fax survey will look:

Hello Mr. Smith, Do you still live in London?	Hello Mr. Jones, Do you still live in New York?
Mr. Smith's Fax	Mr. Jones' Fax

Print Driver

Normal mail merge jobs are sent to a hardcopy printer. Fax merge jobs must be imaged into a single PCL file. This PCL file must contain all the individualized pages for all fax merge recipients. There are two ways to accomplish this:

Send your word processor's mail merge output directly to a VSI-FAX print driver. This will automatically create a single PCL file that can be sent to the fax server via one of the fax clients.

Use your computer's **Print to File** command to save your word processor's mail merge output to a PCL file. The advantage to this method is that you can use this with various server integration techniques (e.g., **vfx** commands, tag and batch files, autosend, etc.) to submit the PCL file to the fax server.

Delivery Mechanisms

After the PCL file has been created by the print driver, it is ready to be submitted to the fax server for fax merge processing. In order to have the fax server perform special fax merge processing on it, you must deliver the PCL file as a fax attachment in a fax envelope with "faxmerge" as the fax number. There are two basic ways to do this:

If you directed your word processor output to one of our fax client print drivers, a fax form will be displayed and the PCL file will automatically be included as an attachment. All you need to do is enter this in the To field:

[Fax : FAXMERGE]

If you printed your word processor output to a PCL file, you can pass that file to the fax server using various delivery mechanisms (e.g., **vfx** commands, tag file, autosend, etc.). Two tags are required:

```
tfn="faxmerge" fll="my_attachment.pcl"
```

Where `my_attachment.pcl` is the mail merge PCL file created by the print driver.

Fax Server

Anytime the fax server receives a fax with a fax number of “faxmerge,” the fax server immediately knows that this is a fax merge job, not a normal fax request. The fax server assigns this job a group fax ID and begins fax merge processing.

Batch Files

Batch files are similar to tag files (page 18). The primary difference is that they can send entirely different faxes to different recipients. This is done by defining several sections within the batch file:

- A *common section* (containing common information for all fax recipients)
- One or more *recipient sections* (containing personalized information for each fax recipient)

File Structure and Syntax

Consider a simple batch file:

```
fnm=MyName
fco=MyCompany
@+COMMON

tfn=5551212
tnm="Mr. Smith"
fll=hello.txt
@+END

tfn=5551213
tnm="Mr. Jones"
fll=goodbye.txt
@+END
```

Common entries always appear at the beginning of the file and continue thru the @+COMMON entry. These two entries (fnm=MyName and fco=MyCompany) define the sender’s name and company, respectively. A common section is optional.

The rest of our sample batch file contains two recipient sections, one for Mr. Smith and one for Mr. Jones. Each recipient section defines a different fax number, fax name and file attachment. Notice that each recipient section ends with a @+END entry.

Delivery and Processing

There can be only one batch file per fax request (e.g., **vfx** command).

All other command line arguments and tag files are processed before the batch file is processed. Therefore, any options or tags following the **-B** option will actually appear in the fax request tag list before any tags in the batch file.

Files referenced in the batch file are processed before files referenced on the command line (including any files referenced in any tag file).

Fax envelopes are created for each recipient in the following order:

- Tag list from the command line
- Tag list from the common section
- Tag list from that recipient section
- Files referenced in the batch file
- Files referenced on the command line

If the last **@+END** entry is missing from a batch file, one is assumed.

Group Files

Often, when you are sending a group fax, your recipient information is actually stored in some sort of database. Most databases support some sort of data export to a neutral file format, typically some form of “delimited ASCII.”

The VSI-FAX group mechanism allows you to pass these delimited ASCII files to the fax server. The fax server will use recipient information in the file to send the group fax.

File Structure and Syntax

Group files have a very simple structure. Consider the following sample group file:

```
Delimiter= ,  
Tagnames=tnm, tfn  
  
Mr. Smith, 5551212  
Mr. Jones, 5551213
```

The first line contains a delimiter statement. The default delimiter is the pipe “|” character. If your database exports to “pipe delimited ASCII” format, you do not need to add this line to your group file.

The second line contains a tag listing. This defines the tag order that will be used to retrieve information from the remaining records in this file. In this example, two tags are used: to name (**tnm**) and fax number (**tfn**).

The remaining lines in the group file contain the name and fax number for each recipient of this group fax. In this case, there are two recipients: Mr. Smith and Mr. Jones.

Delivery and Processing

Group files must be identified to the fax server using the **vfx -g <local_grp_file>** or **-G <srvr_grp_file>** options. The **-g** option is used when the group file is on your local hard drive; the **-G** option is used when the group file is located on the fax server in the `$VSIFAX/lib/groups` directory.

Fax Merge Using Group and Template Files

Although template files are primarily used to customize various email messages sent by the fax server, they work especially well with group files. The combination of the two is ideally suited for sending highly personalized faxes to large groups.

You can use the group file to define various personalized pieces of information for each recipient (this information is typically retrieved from a database and exported to some delimited ASCII text file).

You can then use a template file to merge the information from the group file into a common fax form.

IMPORTANT: When you send a group fax using a template file, the template file must be the first file attachment defined in the fax envelope. Otherwise, it will be imaged as a plain ASCII text file.

Template File Delivery and Processing

Because template files can be plain ASCII text, PostScript or PCL files, all of which can be natively imaged for faxing, the fax server must have some way of knowing when to perform the special processing needed to interpret tag statements inside a template file. This is done by specifying “tag processing” with the **vfx -t fcv=tags** option. This can be done using the **vfx** command line, in a tag or batch file or by using embedded commands.

Consider again the simple group file `groups.txt` from one of our previous examples:

```
Delimiter= ,
Tagnames=tnm, tfn
```

```
Mr. Smith, 5551212
Mr. Jones, 5551213
```

Consider also the example template file:

```
Dear ${tnm:Sir},
```

```
A fax was sent to ${tfm} concerning ${sub:your recent order}.
```

Sincerely,
\${fnm:VSI-FAX}

To send these files to the fax server, enter:

```
vfx -g groups.txt -t fl1=confirm.txt -t fcv=tags
```

This is how each person's fax will look:

Dear Smith,

A fax was sent to 5551212
concerning your recent order.

Sincerely,
VSI-FAX

Mr. Smith's Fax

Dear Jones,

A fax was sent to 5551213
concerning your recent order.

Sincerely,
VSI-FAX

Mr. Jones' Fax

Notice that the actual names and fax numbers were used from the group file. Notice also that because the group file did not specify a subject (i.e., **sub** tag) or send name (i.e., **fnm** tag) that the default subject “your recent order” and send name “VSI-FAX” were used, respectively.

FORMS OVERLAY

You can specify forms to be overlaid onto selected pages of the resultant fax file. These forms must be a TIFF file. More than one overlay file can be specified.

A form file is specified using the tag **ovl=<form_name>** or **ovs=<form_name>**, depending on whether the overlay form is local or on the server. An optional page range can be specified, which will indicate onto which pages of the resultant fax file the form is to be overlaid. If no page range is specified, the form will overlay all pages in the resultant fax file. The page range is specified with the **fpg=<page_range>** tag. The first page of the form is the only page used in the overlay process. This page is overlaid on all specified pages in the resultant fax file.

For example, a letterhead could be overlaid onto the first page of a fax with this command:

```
vfx ... -t ovl=letterhd.tif -t fpg=1 ...
```

An invoice form could be overlaid onto all pages except the first one with this command:

```
vfx ... -t ovl=invoice.tif -t fpg=2- ...
```

You could overlay both a letterhead and an invoice form with the following command:

```
vfx ... -t ovl=letterhd.tif -t fpg=1 -t ovl=invoice.tif -t fpg=2 ...
```

NOTE: An overlay form can be specified in a tag or batch file (similar to all other VSI-FAX tags).

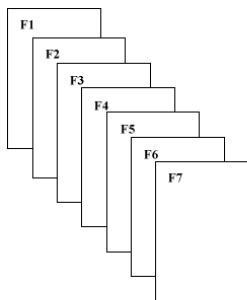
The management of overlay forms can be simplified by combining them into a single multi-page file. In this way, each page of the file corresponds to a particular overlay form (such as a purchase order, invoice or past due notice) that can then be referenced by its page number. This eliminates the need to manage multiple overlay files and allows integrators to maintain a consistent reference when specifying which overlay form to use. Furthermore, changes can be made to an existing forms file without requiring it be rebuilt from scratch, as long as it maintains the same page order sequence. By consolidating the overlay forms into a single forms file, you have the consistency of a known filename and the flexibility to specify the overlay page number (or numbers) to be used on the target document.

Now you are ready to integrate the overlay utility into your application's VSI-FAX interface. You can either specify that a single form be overlaid onto multiple pages of the target file, that multiple overlays be overlaid onto multiple pages or that a single overlay be overlaid onto a page range or selected pages.

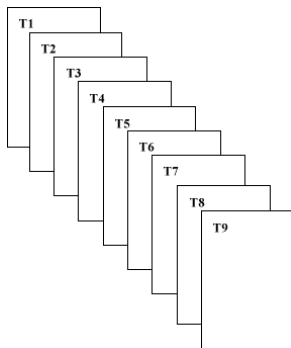
Single Overlay

This example begins with two files: form overlay `form1.tif` (pages F1-7) and text overlay `target.tif` (pages T1-9).

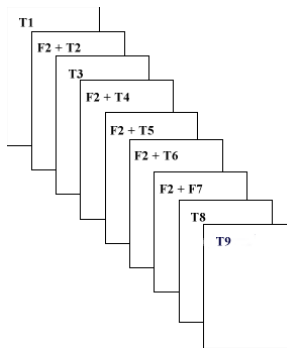
`Form1.tif`



`target.tif`



`result.tif`



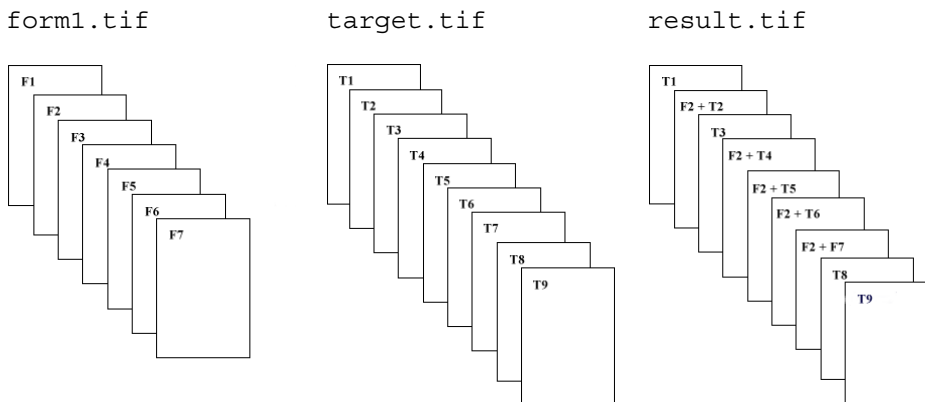
The `result.tif` output file is created by entering the following on a single line:

```
vtifftool overlay -E fine -o result.tif form1.tif:2 target.tif:2,4-7
```

Notice that page 2 of `form1.tif` has been overlaid onto pages 2, 4, 5, 6 and 7 of `target.tif` to produce the `result.tif` output file. When this file is faxed, the fax will appear as if the form overlay and the text overlay are one a single image.

Multiple Overlays

To perform multiple overlays from both `forms1.tif` and `target.tif` to `result.tif`, type the following:



The `result.tif` output file is created by entering the following on a single line:

```
vtiff tool overlay -E fine -o - forms1.tif:2 target.tif:2,4-6 |  
vtiff tool overlay -E fine -o result.tif forms1.tif:5 -:7,9
```

Because the `vtiff tool` overlay utility can only process one overlay operation at a time, the first execution of `vtiff tool` uses a `-` character as the argument to its output switch (`-o`). This redirects output to the standard output device (stdout). The output is then piped to the second execution of the `vtiff tool`, which reads it in as standard input (stdin), performs the overlay operation onto it, then writes its output to the file `result.tif`.

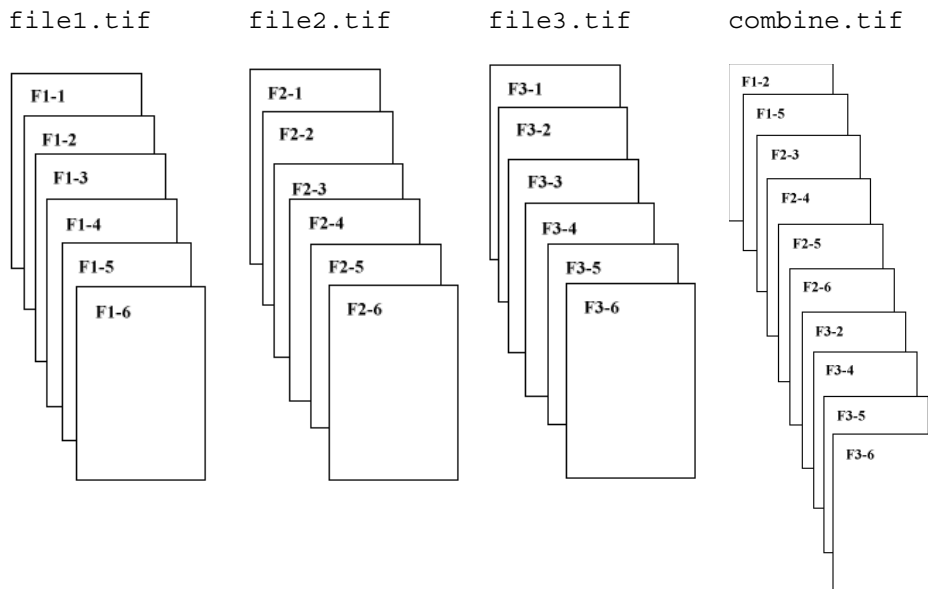
In this example, the `-` character in the final `vtiff tool` argument (`-:7,9`) tells `vtiff tool` to read from the standard input device (stdin) instead of a file.

The result of this procedure is that page 2 of `forms1.tif` has been overlaid onto pages 2, 4, 5, and 6 of the `target.tif`. The output of this operation has then been piped to the second procedure, in which page 5 of `forms1.tif` has been overlaid onto its standard input to create the output file `result.tif`. The original `target.tif` and `forms1.tif` are left in their original state and can therefore be reused later.

Selective Merging

Under certain circumstances, you may need to extract selected pages from one received fax file and merge them with a second file to create a new output file that can then be faxed from VSI-FAX. In this instance you would use the merge option of the `vtifftool` utility, which will allow you to select specific pages from multiple files and merge them into a single document.

This example begins with three multi-page files, `file1.tif` (page F1), `file2.tif` (page F2) and `file3.tif` (page F3). A new file `combine.tif` will comprise pages 2 and 5 from `file1.tif`, pages 3 through 6 from `file2.tif` and pages 2, 4, 5, and 6 from `file3.tif`.



The `combine.tif` output file is created by entering the following on a single line:

```
vtifftool merge -E fine -o combine.tif file1.tif:2,5 file2.tif:3-6  
file3.tif:2,4-6
```

Notice that `combine.tif` has been created, which contains pages from `file1.tif`, `file2.tif` and `file3.tif` arranged in the order that was specified. The `combine.tif` file can now be faxed with VSI-FAX like any other faxable file.

Creating Electronic Business Forms

Before you begin converting your business forms into electronic image files, you need to organize these forms in a page sequence for later reference. You can either scan or fax the forms to create the overlays. Each method is discussed below.

NOTE: You must consolidate all your forms into a single file and maintain the original page order as they are converted to a TIFF image file.

Manual Fax Scanning

Collect your forms together in the proper page sequence, and fax them to your fax server. View the received fax file to confirm that each page is intact and aligned correctly. You can expect a slight shrinkage (about 1/6 inch) of the scanned image versus the original document due to the scan line rendering. This is a normal occurrence with any faxed document. The slight shrinkage should not matter. Because the size of the text overlays will be in proportion to the original document.

You may need to make small positioning adjustments to the text before it overlays exactly onto your imaged form. You can do this with either PCL5e (Hewlett Packard LaserJet) or PostScript (Level 1) codes that use both horizontal and vertical positioning.

The final step is to remove the fax header that was added to each page during transmission. To do this:

1. Log in as root or vsifax and set the VSI-FAX environment.

NOTE: This example assumes that you renamed the received form file as `form.tif` and that this file is located in (`$VSIFAX/faxq/vsifax`), which is the Fax Box directory of user vsifax.

2. Use the **vtifftool** utility to remove the fax header from the multi-page `form.tif` file as follows:

```
cd $VSIFAX/faxq/vsifax
vtifftool clear -E fine -m 0.3i -o overlay.tif form.tif
```

The purpose of `-m 0.3i` is to instruct vtifftool to clear 0.3 inches from the top of the page. You may have to reduce this amount to 0.2 inches (`-m 0.2i`) depending on the output of your particular fax machine.

3. Complete

Scanner Support

You can also use a document scanner to combine forms into a single TIFF image file that can be saved on the VSI-FAX system. Keep in mind that the TIFF file type must be set to one supported by VSI-FAX, as defined by these parameters:

Compression	CCITT G3 (required) Aldus G3 Mac PacBits uncompressed
Maximum Resolution	200x200 dpi
Imaging	Black and White (No Grayscale Support)

Creating Faxable Signature and Logo Files

VSI-FAX supports both shared and private logo and signature files. Files that you want to include in a document must reside in one of two locations:

Public (globally available) files are stored in the `$VSI_FAX/lib/images` directory.

Private (accessible only by the owning user) files are stored on the user's local hard drive.

To create a signature or logo file, follow these steps:

1. Do one of the following:

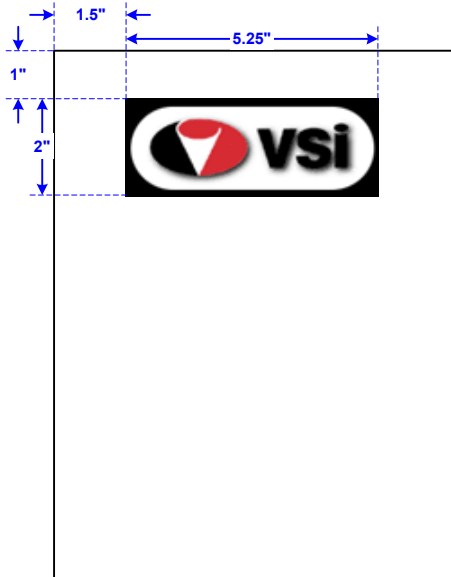
IF	DO THIS
Creating a signature file	Write your signature in black ink on a blank piece of white paper. The signature should be no less than one inch from the top and one inch from the left of the page.
Creating a logo file	Copy your logo onto a faxable piece of paper so that the logo is no less than one inch from the top and one inch from the left of the page.

2. Fax this page to your VSI-FAX user account from a physical fax machine. Make sure you use fine resolution mode in order to obtain the best possible image.

NOTE: By default, all received faxes will be stored in user vsifax's inbox (`$VSI_FAX/faxq/vsifax`) in TIFF file format.

3. Open the received TIFF file and measure the x and y coordinates of the upper-left corner of the logo or signature, as well as the image area width and height.

For example, consider the following scanned image file (`rawlogo.tif`) containing a company logo:



Notice that the actual company logo is 5.25" wide and 2" high. The logo image area is offset 1.5" horizontally (i.e., x-axis) and 1" vertically (i.e., y-axis) from the upper left hand corner of the page.

4. Create a faxable signature or logo file using the **vtifftool cut** command.

For example, to create a faxable logo file (`logo.tif`) by cropping all unused white space from the previous example, enter the following on a single line:

```
vtifftool cut -E fine -x 1.5i -y 1.0i -w 5.25i -h 2.0i -o logo.tif  
rawlogo.tif
```

The output file contains only the signature, so the content of the file is only 5.25" wide and 2" high. All unused white space and additional file information has been removed.

5. To make this a public image file (available to all VSI-FAX users), upload it to the `$VSI/FAX/lib/images` directory using the **vupload image** command.

For example, to upload the logo file from the previous example, enter:

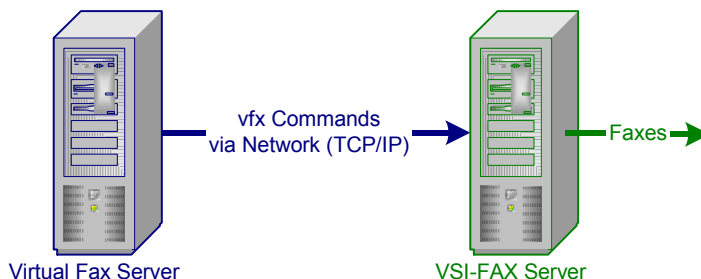
```
vupload image logo.tif
```

You are now ready to include your signature or logo file in a faxable document.

6. Complete.

VIRTUAL FAX SERVER

The virtual fax server is a remote **vfx** command line client. It includes all the functionality of a full fax server except that you cannot connect fax devices to it or send faxes directly from it. You must connect your virtual fax server to a full fax server, via a network, to complete your integration.



Do I Need This? If you intend to use **vfx** commands as your delivery mechanism (page 4) and you need to issue these commands remotely (i.e., from a different IP address than the fax server IP address), you must install a virtual fax server on the remote computer or workstation.

Capabilities and Limitations

Once the virtual fax server is installed, you can issue **vfx** commands just as if you were logged directly into the fax server. In fact, you have access to all the commands and utilities available in a full fax server (e.g., administrative commands and reporting utilities).

The only real limitation is that you cannot connect fax devices directly to a virtual fax server.

Requirements

Virtual fax servers can be installed on any supported platform (e.g., Linux, Unix or Windows NT/2000).

Virtual fax servers use a special “Virtual Fax Server” license, which you must obtain from your VSI authorized reseller or directly from VSI before you can install it.

In order to send or receive faxes, the virtual fax server and the full fax server must share the same local area or wide area network.

Refer to your *Getting Started Manual* for virtual fax server installation instructions.

LP SPOOLER

The **vfx** command is modeled after the Unix **lp** spooler command. The similarities between the two allow you to submit a fax as easily as printing a document.

Integration is accomplished by using an lp-compatible printer interface script. All Unix and Linux operating systems, with the exception of IBM AIX, support ASCII text printer interface scripts. IBM AIX systems use a database architecture that requires their printer interface scripts to be compiled. For AIX IBM interface scripts, refer to the IBM AIX documentation for additional information about compiling interface scripts.

Creating a Printer Destination for Faxing

In order to use the lp spooler for fax interrogations, you need to create a printer destination called “fax.” The name can be anything that works for your environment. This printer destination is created the same way a regular printer would be created. The fax printer that you create should have a device port of `/dev/null` and a printer type of ASCII (or another generic type).

When you create the fax printer destination, the operating system automatically puts a default printer script for that printer in the interfaces directory. On most, but not all Unix and Linux systems, the interfaces directory is:

```
/usr/spool/lp/admins/lp/interfaces
```

Refer to your operating system documentation for additional information about adding an ASCII printer to your system.

Creating a Custom Printer Interface Script

Once you have successfully added the new fax printer destination to your system, you need to replace or modify that default interface script to suit your particular environment and fax requirements.

The most common method of lp faxing is to use the Unix **lp** command. Typically, the **lp** command accepts several command line options that work well for sending a fax. In order for lp faxing to work correctly, two arguments must be passed to the interface script:

- Fax number
- Name of the file you want to fax

The Unix **lp** command allows the use of the **-o** argument, which we will use to specify the fax number. The filename can either be explicitly listed or a file can be redirected to the **lp** command. Either method works for lp faxing.

When creating the interface script. The most important line is the **vfxx** command line. This is where various options can be added to make the lp faxing work in your environment. The sample script (page 51) only uses a few options (login name, fax number and the file name). Additional options could be added that would include a cover page and use a user's default profile settings to send the fax. All command line options available to the **vfxx** command are also available within the interface script.

Refer to the *Appendix A – vfx Command Reference* (page 67) for additional information about the **vfxx** command.

Sample Interface Script

This sample interface script replaces the default script used by the fax printer destination. It allows users to specify the fax number on the **lp** command line. This allows the specified document to be faxed.

IMPORTANT: The sample script that is shown here is a sample. It will not work in every possible instance. It does demonstrate how and what an interface script should do.

When using the Unix **lp** command for faxing, a command line would typically be something like:

```
lp -d fax -o <fax_number> <file_name>
```

For example, to fax the `/etc/motd` file to fax number 111-222-3333, enter:

```
lp -d fax -o 111-222-3333 /etc/motd
```

This sample script (`faxfilter`) is provided in your `/samples` directory.

```
# Set the VSI environment
. /etc/vsifax3.sh
# Store the login name of the submitter
LOGIN=$2
# Extract the fax number
FAXNUM="$5"
# Extract the file to be faxed
shift 5
FILES="$*"
# Submit the fax request
vfx -U $LOGIN -n "$FAXNUM" -E fine $FILES 2>&1 | mailx -s <subject>
$LOGIN
# Exit script
exit 0
```

NOTE: In the previous example script, `<subject>` is any user-defined character string you want for your email subject line.

WORDPERFECT INTEGRATIONS

Because many of our customers want to fax enable WordPerfect on Unix, we have created two integrations using WordPerfect macros and our fax print driver.

X Windows Versions

This VSI integration for WordPerfect 7/8 is only designed to work with graphical X Windows versions of WordPerfect. Also earlier versions of WordPerfect may work but have not been tested.

Files

The graphical integration consists of these files.

<code>faxwp.x.sh</code>	Script that builds a UI and faxes document.
<code>vsipclx.prs</code>	Print driver for WordPerfect.

Configuring the Integration

In order to use this integration, perform this configuration procedure:

1. Login as root. (or **su** to root).
2. Change directory to the WordPerfect `/shbin10` directory.
3. Execute the **xwpdest** program to create a new printer destination.

The printer destination must have a custom spool command. The command must be:

```
$VSI/FAX/lib/wp/faxwp.x.sh <F> pcl
```

4. Save this destination and exit **xwpdest**.

5. Create a printer. The printer can either be a system printer or a personal printer. It must be a PCL type printer. Either the `$VSI/FAX/lib/wp/vsipclx.prs` driver or the default `hplj.prs` file can be used.

When setting up the printer it must use the VSI destination that was created in steps 3 thru 4.

Once the printer is setup, the integration is ready to be used.

6. Complete.

Using the Integration

To fax a document from within WordPerfect, simply print the document to the VSI fax print driver. After the document is converted to PCL format, a dialog box will open and allow you to enter the recipient information.

Character-Based Versions

This VSI integration for WordPerfect 7/8 is only designed to work with non-graphical (character-based) versions of WordPerfect. Also earlier versions of WordPerfect may work but have not been tested.

Files

The non-graphical integration consists of these files.

<code>vsi.wpm</code>	WordPerfect macro that calls the imaging and faxing script.
<code>faxwpc.sh</code>	Script used for imaging and faxing.
<code>vsipcl.prs</code>	PCL print driver for WordPerfect.
<code>wpc.config</code>	Configuration for defining <code>vsiwp.sh</code> .
<code>vsiwp.env</code>	Environment settings needed by <code>faxwpc.sh</code> ; created from the <code>wpc.config</code> script.

Configuring the Integration

1. Login as root (or **su** to root).

You must know the full path to the WordPerfect installation directory.

2. Change directory to `$VSI/FAX/lib/wp`.
3. Execute the **wpc.config** script.

4. When prompted, enter the path to the WordPerfect installation.
5. Complete.

Using the Integration

To fax the current document from within WordPerfect, simply execute the “vsi” macro as follows:

1. Choose **Tools > Macro > Execute**.
2. Enter **vsi** and press **ENTER**.

You will then be prompted to enter fax recipient information. You will also be able to select a cover page from the currently installed cover pages.

3. After entering the fax recipient information, you will be given a fax request ID for tracking this request.
4. Complete.

Notes

You must save your file before you can fax it.

Some WordPerfect installations use the "fast-save" feature. This is not compatible with this integration. If errors are shown about fast-saved files. You must turn off this feature.

This integration uses the **wprntc** and **wprnt7c** utilities to convert the WordPerfect document to PCL format. In some cases the WP7 **wprnt7c** utility returns an `invalid port specified` message. If this occurs, the `faxwpdoc.sh` script can be modified to correct this problem. The script can be changed to use the WordPerfect **wprnt** command instead of **wprnt7c**. The original file includes a likely syntax for this fix.

The **wprntc** and **wprnt7c** programs can cause the WordPerfect print spooler to improperly handle some documents. This can often occur when the "fast-save" feature is turned on. If the `Printer action required` message appears, this usually indicates that some job is stuck in the print spooler. To clear the print spooler:

1. Cancel any pending jobs using the **Printer** menu.
2. Close WordPerfect.
3. Remove the `$HOME/.wprc` directory.
4. Restart WordPerfect.
5. Retry your fax.

This integration only works with non-graphical (character-based) WordPerfect versions 7.x and 8.x. Earlier versions of WordPerfect might work. If you are trying to use an earlier version WordPerfect, you will probably need to modify the `faxwpdoc.sh` script to use the proper command for converting the file.

USING XML-F

Extensible Markup Language (XML) comprises various Data Type Definitions (DTDs). Each DTD is a set of tags for describing a particular collection of data. In XML, one DTD is created for each collection of data that requires description. For example, to create an XML language for Accounts Payable, one DTD would be required to describe the vendors, one to describe the purchase orders, and another to describe invoices, etc.

A collection of data is generally accompanied by its DTD whenever it is stored or transferred from one place to another, creating “documents that know themselves.” The inclusion of the DTD means that the content or meaning of the data is embedded in the document in a way that is human-legible and easily interpreted by any XML enabled software.

VSI has gone one step further and has proposed a set of DTDs for network fax transactions called XML-F. This interface is a simple, powerful way to pass fax transactions to and from fax servers. It is based on four simple structures that allow any conforming application to:

- Send a fax
- Get the status of a sent fax
- Cancel a fax
- Get fax resources

XML-F uses eight XML document types to implement these four features: four requests and four responses. Each feature has a corresponding request and response document.

For clarity, the examples in the remainder of this chapter use simplified XML-F document types. Refer to *Appendix B – XML-F DTDs* (page 73) for complete DTD listings.

The *xmlf* Command

Synopsis

```
$VSIFAX/bin/xmlf [-h <URL>] [-o <file>] [-t {html | text | xml}]  
  <file> [<attachment>]
```

Description

xmlf is the transport that sends a fax file to the server for transmission.

Options

-h <URL>	Fax server Universal Resource Locator (URL). Default is local host.
-o <file>	The name of the file to put the response from the send operation. Default is displaying the response on the screen.
-t {html text xml}	Response format.
<file>	The name of the XML file you are sending to the server.
<attachment>	The name of a file you want to add.

Notes

When entering a URL, the following format is required:

```
[transport://] <host> [:port number]
```

Where:

transport:	Defaults to vxml .
<host>	Name of the fax server.
port number	Port number that the fax server uses to send faxes.

Fax-Submit Document

The fax-submit document describes the elements necessary to set up and send a fax on an XML-F compliant system. The fax-submit document has the following basic structure (this example is simplified):

fax-submit	
account	(required)
id	(required)
subid	(zero or one)
mail-address	(zero or one)
recipient	(one or more)
personal-name	(zero or one)
company-name	(zero or one)
fax-number or canonical-fax	(required)
voice-number or canonical-voice	(zero or one)
sender	(required)
personal-name	(required)
company-name	(zero or one)
fax-number or canonical-fax	(zero or one)
voice-number or canonical-voice	(zero or one)
subject	(zero or one)
content	(required)
body	(zero or more)
application-reference	(zero or one)
command-reference	(zero or one)

ELEMENT	DESCRIPTION
account	The account contains the information necessary to identify the user of the service. Authentication is left to the transport and implementation, the account structure contains an ID intended to identify the billing entity and a sub-ID to be used for grouping of departments or users within the billing entity. The account also specifies a mail-address that is an email channel available for the fax service to use for error or other messages.
recipient	The recipient defines one or more people for whom this fax transmission is ultimately intended. A fax submission must have at least one recipient, however, the recipient need only have fax-number defined.
sender	The sender entry is required, with at least the personal name entered for a sender.
subject	The subject is optional.

ELEMENT	DESCRIPTION
content	The content element contains zero or more body elements. Body elements must contain a filename attribute and may specify associated content-type (RFC 2046) and content-transfer-encoding (base64 or none).
application reference	An application-reference tag is defined to permit the submitting application to provide an arbitrary application-specific reference to this fax request. This reference tag may be used to request status information on this fax submission.

Finally, in XML-F, all requests carry a command-reference, which is intended for the sending system to uniquely identify the particular request. This is useful for debugging and matching up particular responses with particular requests when the system is used asynchronously.

The following is an example of a completed send document:

```
<?xml version="1.0"?>
<!DOCTYPE fax-submit SYSTEM "fax-submit.dtd">
<fax-submit resolution="fine" coversheet="yes">
  <account>
    <id>
      Filmore5455
    </id>
    <subid>
      David Filmore
    </subid>
    <mail-address>
      filmore@vsi.com
    </mail-address>
  </account>
  <email-notification when="on-success">
    filemore@vsi.com
  </email-notification>
  <sender>
    <personal-name>
      David Filmore
    </personal-name>
    <company-name>
      Willy Wonka Chocolates, Inc.
    </company-name>
    <fax-number>
      237-0998
    </fax-number>
    <voice-number>
      238-9873
    </voice-number>
  </sender>
</fax-submit>
```

```

    <recipient>
      <personal-name>
        Rob Juergens
      </personal-name>
      <canonical-fax>
        <country-code>
          011
        </country-code>
        <area-code>
          987
        </area-code>
        <local-number>
          242-1234
        </local-number>
      </canonical-fax>
      <voice-number>
        011-987-242-5678
      </voice-number>
    </recipient>
    <subject>
      The best looking XML we are able to
produce...
    </subject>
      <content>
        <body filename="inline.txt">
Some people are destined to discover that when they get to
the end of time, there they are.
        </body>
        <body filename="inline.txt"
          content-
type="text/plain"
          content-transfer-
encoding="base64">
VGhpcyB3YXMgYmFzZTY0IGVuY29kZW
QgPGludmFsaWQ+PC9pbmZhbGlkPg==
        </body>
        <body filename="inline.txt">
This is some text that we think would make for a wonderfully
interesting fax body had anyone had time to actually think up
something fun and interesting to say. Or should anyone ever decide
to read it.
        </body>
      </content>
    </command-reference>

```

```
                                xxs234234s
                                </command-reference>

</fax-submit>
```

Upon successful receipt of a valid fax-submit, a fax-submit-response will be returned.

Fax-Submit-Response Document

Upon receipt of a fax-submit request, an XML-F service should respond by providing a fax-submit-response. This response is used to acknowledge receipt of the fax submission upon parsing and validation of the request. The response is used to provide a server-side reference. It comprises the following structure (this example is simplified):

```
fax-submit-response
    request-results          (required)
    service-reference        (required)
    application-reference    (zero or one)
    command-reference        (zero or one)
```

ELEMENT	DESCRIPTION
request results	The request-results provide information on the results of the fax-submit. Request-results contain attributes to indicate the status of the request (normal, warning, or failed) and the reason for the status. The data attached to the request-results entity provides an ad hoc message back to the application.
service reference	This is a unique reference assigned to a particular fax submission by the service provider. This is the best reference for the user application to use when requesting status or when canceling a request.
application reference	This is the optional (but recommended) reference that the submitting application assigned to the fax request triggering this response. This reference may be used to get status or to cancel the request if the application has not received a fax-submit-response (and therefore no service-reference would be known).

Fax-Status Document

The fax-status document is used by an application to request status on a service about a particular fax request. The fax-status request can be used to request short and detailed status reports in either XML or text-formatted form. As well, the fax-status request can ask the service to mail the resulting report to an email address.

The application can either use the application-reference or the service-reference when referring to this request. This permits the submitting application to issue status requests without having to wait or process a fax-submit-response document prior to asking for status. In the event the fax-submit-response is never received, the application must use the application-reference to get information on the request.

The fax-status comprises the following structure (this example is simplified):

fax-status	
account	(required)
id	(required)
subid	(zero or one)
mail-address	(zero or one)
email-to	(zero or more)
service-reference	(must specify either service
application-reference	or application reference)
command-reference	(zero or one)

This is an example of a completed fax status document:

```
<?xml version="1.0"?>
<!DOCTYPE fax-status SYSTEM "fax-status.dtd">
<fax-status results-format="xml" report-type="full">
    <account>
        <id>
            Filmore5455
        </id>
        <subid>
            David Filmore
        </subid>
        <mail-address>
            filmore@vsi.com
        </mail-address>
    </account>
    <service-reference>
        1011
    </service-reference>
    <command-reference>
        xxs234235s
    </command-reference>
</fax-status>
```

Fax-Status-Response Document

The fax-status-response document provides the detailed information about what happened (or is happening) to a fax request on an XML-F service. The basic structure for the fax-status-response is as follows (this example is simplified):

fax-status-response	
request-results	(required)
status (one of...)	(required)
{short-status}	
job-status	(required)
{full-status}	
job-status	(required)
attempt-status	(one or more)
recipient	(required)
date	(required)
csi	(zero or one)
result	(required)
short-message	(required)
long-message	(zero or one)
service-reference	(required)
application-reference	(zero or one)
command-reference	(zero or one)

Fax-Cancel Document

The fax-cancel document is used by a sending application to cancel a previously submitted fax request. This is a sample fax cancel document:

```
<?xml version="1.0"?>
<!DOCTYPE fax-cancel SYSTEM "fax-cancel.dtd">
<fax-cancel>
    <account>
        <id>
            Filmore5455
        </id>
        <subid>
            David Filmore
        </subid>
        <mail-address>
            filmore@vsi.com
        </mail-address>
    </account>
    <service-reference>
        1011
    </service-reference>
    <command-reference>
        xxs234236s
    </command-reference>
</fax-cancel>
```


APPENDIX A – VFX COMMAND REFERENCE

This is complete documentation for the **vfx** command.

Synopsis

```
vfx [-A <alias>] [-B <file>] [-c <file>] [-C <cover>]
    [-d {<device> | <class>}] [-E {std | fine}]
    [-F {ep | fax | pcl | ps | tif | txt}] [-g <srvr_grp_file>]
    [-G <local_grp_file>] [-H <host>] [-i] [-l {letter | a4 | legal}]
    [-L {attachments | covers | dests | folders | images | overlays
    | retrys}] [-m {both | each | fail | none | ok}] [-M <email>]
    -n <fax_num> [-N] [-o <file>] [-p {l | m | h | u}] [-r <retry>]
    [-R] [-s] [-S {on | off}] [-t <tag>=<value>] [-T <time>] [-U <user>]
    [-x] <file>
```

Description

vfx provides a full-featured command-line interface for sending faxes.

You can include many different kinds of file attachments with your fax. However, the exact types supported by any given fax server is highly dependent on that particular operating system and environment. Refer to your *VSI-FAX Server Manual* for a detailed discussion of various file types and how they are imaged on various platforms.

vfx submits a fax envelope to the fax server, which comprises the following:

- Zero or more files or folders to be imaged (a fax can consist of a cover page only)
- Zero or more forms to be overlaid on selected pages of the resultant fax file
- One or more recipients to send the faxes to
- A set of cover page parameters
- A set of send parameters (e.g., priority, send time, etc.)

vfx verifies all parameters given to it, in order to find as many errors as possible before actually submitting the request to the server. Therefore, if **vfx** is successful in submitting the request to the server, it is likely that the request will actually be faxed.

One design parameter of **vfx** is that the user is guaranteed to be able to delete any files passed to it immediately after the **vfx** command returns. **vfx** will copy a file if necessary to avoid requiring any reference to the file after it returns.

The output of the **vfx** command is a fax request ID, which can be used to track the status of the fax request.

Options

-A	<alias>	Directory (phone book) person alias to send to.
-B	<file>	Batch <file> to process.
-c	<file>	Tag <file> to process.
-C	<cover>	Cover page to include with this fax.
-d	{<device> <class>}	Device or class used to send this fax. Default is system default fax device or class.
-E	{std fine}	Send resolution. Valid values are: <div> <div>std</div> <div>Standard (204 x 98)</div> </div> <div> <div>fine</div> <div>Fine (204 x 196) (default).</div> </div>
-F	<extension>	File type <extension> . Valid values are: <div> <div>txt</div> <div>Text file (default).</div> </div> <div> <div>tif</div> <div>TIFF file.</div> </div> <div> <div>fax</div> <div>VSI 2.x fax file.</div> </div> <div> <div>pcl</div> <div>PCL file.</div> </div> <div> <div>ps</div> <div>PostScript file.</div> </div> <div> <div>ep</div> <div>Epson file.</div> </div>
-g	<srvr_grp_file>	Group file in the \$VSI_FAX/lib/groups directory.
-G	<local_grp_file>	Directory (phone book) group or local group file to send to.
-H	<host>	Connect to this fax server host name.
-i		Interactive cover page tag entry.
-l	<length>	Page <length> . Valid values are: <div> <div>letter</div> <div>11 inches (default).</div> </div>

	a4	11.69 inches.
	legal	14 inches.
-L	List resources. Valid resource types are:	
	attachments	List of attachments.
	covers	List of cover pages.
	dests	List of fax devices and classes.
	folders	List of folders.
	images	List of images.
	groups	List of groups.
	overlays	List of overlays.
	retrys	List of retry methods.
-m	<mode>	Email notify <mode> . Valid mode values are:
	both	Email always sent.
	each	Email sent for each attempt.
	fail	Email sent if failed (default).
	none	Email never sent.
	ok	Email sent if successful.
-M	<email>	<email> address to send notifications to. If not supplied, default is to retrieve the email address from the user profile.
-n	<fax_num>	Phone number to send to.
-N		Scan file for phone number.
-o	<file>	Image request and output to <file> .
-O		Send cover page only.
-p	{l m h u}	Priority. Valid values are:
	l	Low.
	m	Medium (default).
	h	High.
	u	Urgent.

-r <retry>	Retry method. Default is default .
-R	Recover saved jobs.
-s	Silent mode.
-S {on off}	Enable scanning of stdin for tags. Default is on .
-t <tag>=<value>	Include this <tag> set to this <value> with this fax. TIP: Enter vfx -help tags or refer to your <i>VSI-FAX Server Manual</i> for a list of valid fax envelope tags.
-T <time>	Send <time> in [{yy YYYY}] [mm] [dd] hhmm [{am pm}] format. Default is send immediately.
-U <user>	User name to login as when sending this fax.
-x	Process XML <file> .
<file>	File to process (fax).

Notes

You can enter overrides for the default values for **<user>** and **<host>**. These can be set in the shell environment, which will then override the values in `$HOME/.vsifax/vsifax.ini`.

The system will give you a `Permission Denied` error if you attempt to send a file that you do not have permission to read.

The **-F** option can be used to include other file types. For example, Windows fax servers can recognize any file extension with a Dynamic Data Exchange (DDE) print entry in the registration database. Refer to your *VSI-FAX Server Manual* for a detailed discussion of various file types and how they are imaged on various platforms.

TIP: You can quickly find out which file types are recognized by your fax server using the **vinfo filetypes** command.

The **-i** (interactive) option is used in conjunction with a cover page. This option will then prompt the user interactively for all tags which the cover page uses, such as “to name,” “to company,” etc. If more than one recipient is specified on the command line, the tags for each recipient will be prompted for in sequence.

The **-L** option tells **vfx** to output to stdout a list of all available resources of the type requested. This list will contain the name and description of each resource available.

If the **-L covers** option is used with the option **-t tags=list**, then a list of available cover pages containing the specified list of tags will be output. Therefore, the following command will give a list of all available cover pages:

```
vfx -L covers
```

The following command lists all available cover pages containing the **fnm** (from name) and **fvn** (from voice number) tags:

```
vfx -L covers -t tags=fnm, fvn
```

The **-n** option is unique in that it can be used to specify more than one tag value. The full string used is:

```
-n <fax_num>:<name>:<company>:<voice_num>:<info>
```

For example, consider the following **vfx** command:

```
vfx -n "555-1212:Joe Smith:Ace Financial Services"...
```

- is equivalent to -

```
vfx -n 555-1212 -t tnm="Joe Smith"  
-t tco="Ace Financial Services"...
```

- or -

```
vfx -t tfn="555-1212" -t tnm="Joe Smith"  
-t tco="Ace Financial Services"...
```

The **-o <file>** tells **vfx** to not submit the envelope for faxing, but to convert it to an image file and return that file to the user. When using this mode, a recipient is not required. (A recipient is obviously needed if you are faxing.) If a cover page is specified, it will be the first page of the image. If more than one recipient is specified, the first recipient will be used to create the cover page. The returned file will be a standard TIFF file. If the file name is "-", the file is written to stdout.

The **-O** (cover page only) option specifies that the fax submittal will consist of a cover page only. If this option is not specified, then at least one file must be sent, and **vfx** will read stdin for input if no files are specified. Usually this option is used with a cover page that supports a note file to supply the message the user wants to send.

The **-s** (silent) option tells **vfx** not to output the Request ID for the job. This option is typically used when the call to **vfx** is embedded in user's application. This will result in a "blind" submittal, since this is the only method of getting the Request ID of the job submitted.

The **-S** (scan) option enables or disables the scanning of stdin for embedded tags. Note that embedded tags are only used if the file comes from stdin. Default is to scan.

Times, entered with the **-T** option, can be entered with either two-digit or four-digit year values.

The **-u** (use user configuration file) option tells **vfx** to load any specified options in the user's home **vsifax.ini** file. This is not done by default in order that a user's own send options can not interfere with the options used in a fax-integrated application.

APPENDIX B – XML-F DTDs

Fax Submit DTD

```
<?xml version="1.0" encoding="UTF-8"?>

<!-- ***** -->
<!-- XML-F Fax-Submit DTD -->
<!-- -->
<!-- Revision: 0.4 -->
<!-- Date: 06/22/99 -->
<!-- -->
<!-- Used for submitting a fax to an XML-F-conformant server. -->
<!-- Server will respond with a fax-submit-response. -->
<!-- ***** -->

<!ELEMENT fax-submit
    (account,
     recipient+,
     sender,
     email-notification?,
     subject?,
     content,
     application-reference?,
     command-reference?)>
<!ATTLIST fax-submit                response-format
    (xml |
     html |
     text)
    "xml">
<!ATTLIST fax-submit                resolution
    (fine |
```

```
        std)
        "fine">
<!ATTLIST fax-submit                priority
    (low |
      normal |
      high)
    "normal">
<!ATTLIST fax-submit                coversheet
    (CDATA)
    "yes">
<!ELEMENT account
    (id,
      subid?,
      mail-address?)>
<!ELEMENT id                        (#PCDATA)>
<!ELEMENT subid                     (#PCDATA)>
<!ELEMENT mail-address              (#PCDATA)>
<!ELEMENT recipient
    (personal-name?,
      company-name?,
      (fax-number |
        canonical-fax),
      (voice-number |
        canonical-voice)?)>
<!ELEMENT personal-name             (#PCDATA)>
<!ELEMENT company-name              (#PCDATA)>
<!ELEMENT fax-number                (#PCDATA)>
<!ELEMENT canonical-fax
    (country-code?,
      area-code?,
      local-number,
      extension?)>
<!ELEMENT voice-number              (#PCDATA)>
<!ELEMENT canonical-voice
    (country-code?,
      area-code?,
      local-number,
      extension?)>
<!ELEMENT country-code              (#PCDATA)>
<!ELEMENT area-code                 (#PCDATA)>
<!ELEMENT local-number              (#PCDATA)>
<!ELEMENT extension                 (#PCDATA)>
<!ELEMENT sender
    (personal-name,
      company-name?,
      (fax-number |
```

canonical-fax)?,	
(voice-number	
canonical-voice)?)>	
<!ELEMENT email-notification	(#PCDATA)>
<!ATTLIST email-notification	when
(on-failure	
always	
on-success	
none)	
"on-failure">	
<!ELEMENT subject	(#PCDATA)>
<!ELEMENT content	(body*)>
<!ELEMENT body	(#PCDATA)>
<!ATTLIST body	filename
(CDATA)	
#REQUIRED>	
<!ATTLIST body	content-transfer-encoding
(base64	
none)	
#IMPLIED>	
<!ATTLIST body	content-type
(CDATA)	
#IMPLIED>	
<!ELEMENT application-reference	(#PCDATA)>
<!ELEMENT command-reference	(#PCDATA)>

Fax Submit Response DTD

```

<?xml version="1.0" encoding="UTF-8"?>

<!-- ***** -->
<!-- XML-F Fax-Submit Response DTD -->
<!-- -->
<!-- Revision: 0.3 -->
<!-- Date: 11/12/98 -->
<!-- -->
<!-- Request response generated upon acceptance of a -->
<!-- fax request. -->
<!-- ***** -->

<!ELEMENT fax-submit-response      (request-results,
    service-reference,
    application-reference?,
    command-reference?)>
<!ELEMENT request-results          (#PCDATA)>
<!ATTLIST request-results          status
    (normal |
     warning |
     failed)
    #REQUIRED>
<!ELEMENT service-reference        (#PCDATA)>
<!ELEMENT application-reference    (#PCDATA)>
<!ELEMENT command-reference        (#PCDATA)>

```

Fax Status DTD

```

<?xml version="1.0" encoding="UTF-8"?>

<!-- ***** -->
<!-- XML-F Fax-Status DTD -->
<!-- -->
<!-- Revision: 0.3 -->
<!-- Date: 11/12/98 -->
<!-- -->
<!-- Document used to request status on a previously -->
<!-- submitted fax request. -->
<!-- ***** -->

<!ELEMENT fax-status
    (account,
     email-to*,
     (service-reference |
      application-reference),
     command-reference?)>
<!ATTLIST fax-status                response-format
    (xml |
     html |
     text)
    "xml">
<!ATTLIST fax-status                report-type
    (short |
     full)
    "short">
<!ELEMENT account
    (id,
     subid?,
     mail-address?)>
<!ELEMENT id                        (#PCDATA)>
<!ELEMENT subid                     (#PCDATA)>
<!ELEMENT mail-address               (#PCDATA)>
<!ELEMENT email-to                   (#PCDATA)>
<!ELEMENT service-referece           (#PCDATA)>
<!ELEMENT application-reference      (#PCDATA)>
<!ELEMENT command-reference          (#PCDATA)>

```

Fax Status Response DTD

```

<?xml version="1.0" encoding="UTF-8"?>

<!-- ***** -->
<!-- XML-F Fax-Status Response DTD -->
<!-- -->
<!-- Revision: 0.3 -->
<!-- Date: 11/12/98 -->
<!-- -->
<!-- Used for reporting the status of a fax request. -->
<!-- ***** -->

<!ELEMENT fax-status-response
    (request-results,
    (short-status |
    full-status),
    service-reference,
    application-reference?,
    command-reference?)>
<!ELEMENT request-results                (#PCDATA)>
<!ATTLIST request-results                status
    (normal |
    warning |
    failed)
    #REQUIRED>
<!ELEMENT short-status                   (job-status)>
<!ELEMENT job-status                     (#PCDATA)>
<!ATTLIST job-status                     current-state
    (new |
    in-progress |
    finished)
    #REQUIRED>
<!ATTLIST job-status                     disposition
    (success |
    partial-success |
    failure)
    #REQUIRED>
<!ELEMENT full-status
    (job-status,
    attempt-status+)>
<!ELEMENT attempt-status
    (recipient,
    date,
    csi?,
    result)>

```



```

<!ELEMENT recipient
    (personal-name?,
     company-name?,
     (fax-number |
      canonical-fax),
     (voice-number |
      canonical-voice)?)>
<!ELEMENT personal-name                (#PCDATA)>
<!ELEMENT company-name                 (#PCDATA)>
<!ELEMENT fax-number                   (#PCDATA)>
<!ELEMENT canonical-fax
    (country-code?,
     area-code?,
     local-number,
     extension?)>
<!ELEMENT voice-number                 (#PCDATA)>
<!ELEMENT canonical-voice
    (country-code?,
     area-code?,
     local-number,
     extension?)>
<!ELEMENT country-code                 (#PCDATA)>
<!ELEMENT area-code                    (#PCDATA)>
<!ELEMENT local-number                 (#PCDATA)>
<!ELEMENT extension                    (#PCDATA)>
<!ELEMENT date                         (#PCDATA)>
<!ELEMENT csi                          (#PCDATA)>
<!ELEMENT result
    (short-message,
     long-message?)>
<!ATTLIST result                        disposition
    (success |
     failure |
     NA)
    #REQUIRED>
<!ELEMENT short-message                (#PCDATA)>
<!ELEMENT long-message                 (#PCDATA)>
<!ELEMENT service-reference            (#PCDATA)>
<!ELEMENT application-reference        (#PCDATA)>
<!ELEMENT command-reference            (#PCDATA)>

```

Fax Cancel DTD

```
<?xml version="1.0" encoding="UTF-8"?>

<!-- ***** -->
<!-- XML-F Fax-Cancel DTD -->
<!-- -->
<!-- Revision: 0.3 -->
<!-- Date: 11/12/98 -->
<!-- -->
<!-- Document used to cancel a previously submitted -->
<!-- fax request. -->
<!-- ***** -->

<!ELEMENT fax-cancel
    (account,
    (application-reference |
    service-reference),
    command-reference?)>
<!ATTLIST fax-cancel                results-format
    (xml |
    html |
    text)
    "xml">
<!ELEMENT account
    (id,
    subid?,
    mail-address?)>
<!ELEMENT id                        (#PCDATA)>
<!ELEMENT subid                     (#PCDATA)>
<!ELEMENT mail-address              (#PCDATA)>
<!ELEMENT service-reference         (#PCDATA)>
<!ELEMENT application-reference     (#PCDATA)>
<!ELEMENT command-reference         (#PCDATA)>
```

Fax Cancel Response DTD

```
<?xml version="1.0" encoding="UTF-8"?>

<!-- ***** -->
<!-- XML-F Fax-Cancel Response DTD -->
<!-- -->
<!-- Revision: 0.3 -->
<!-- Date: 11/12/98 -->
<!-- -->
<!-- Fax-Cancel response document. -->
<!-- ***** -->

<!ELEMENT fax-cancel-response
    (request-results,
     service-reference,
     application-reference?,
     command-reference?)>
<!ELEMENT request-results          (#PCDATA)>
<!ATTLIST request-results          status
    (normal |
     warning |
     failed)
    #REQUIRED>
<!ELEMENT service-reference        (#PCDATA)>
<!ELEMENT application-reference    (#PCDATA)>
<!ELEMENT command-reference        (#PCDATA)>
```

Fax Resource DTD

```
<?xml version="1.0" encoding="UTF-8"?>

<!-- ***** -->
<!-- XML-F Fax-Resource DTD -->
<!-- -->
<!-- Revision: 0.3 -->
<!-- Date: 05/02/99 -->
<!-- -->
<!-- Document used to request a list of resources. -->
<!-- ***** -->

<!ELEMENT fax-resource
    (account,
     resource-type,
     command-reference?)>
<!ATTLIST fax-resource      response-format
    (xml |
     html |
     text)
    "xml">
<!ELEMENT account
    (id,
     subid?,
     mail-address?)>
<!ELEMENT id                (#PCDATA)>
<!ELEMENT subid             (#PCDATA)>
<!ELEMENT mail-address      (#PCDATA)>
<!ATTLIST fax-resource      resource-type
    (priority,
     resolution,
     coversheet)
    #REQUIRED>
<!ELEMENT command-reference  (#PCDATA)>
```

Fax Resource Response DTD

```

<?xml version="1.0" encoding="UTF-8"?>

<!-- ***** -->
<!-- XML-F Fax-Resource Response DTD -->
<!-- -->
<!-- Revision: 0.3 -->
<!-- Date: 05/02/99 -->
<!-- -->
<!-- Fax-Resource response document. -->
<!-- ***** -->

<!ELEMENT fax-resource-response
    (request-results,
     resource-type,
     resource-list,
     command-reference?)>
<!ELEMENT request-type          (#PCDATA)>
<!ATTLIST request-results      status
    (normal |
     warning |
     failed)
    #REQUIRED>
<!ELEMENT command-reference    (#PCDATA)>
<!ELEMENT resource-list
    (entry+)>
<!ELEMENT entry
    (name,
     description)>

```


INDEX

Symbols

- (hyphen) 41
- @+IMAGE command 21–24
- \ (backslash character) 23
- | (pipe character) 36
- ~ (tilde character) 23

A

- account element 59
- AIX systems 49
- application-reference element 60, 62
- applications
 - See also specific applications*
 - database programs 31
 - enabling with vfx tags 17–19
 - integrating fax services in 1–16
 - spreadsheet programs 31
 - Windows applications 14
 - word processor programs 31, 32–34
- ASCII files. *See* text files
- ASCII print driver 12, 25, 26
- ASCII printers 49
- ASCII text printer interface scripts 49
- attachments 2, 3, 34, 67
- authentication 59
- autofax account 27–30
- autosend directory 9–10

B

- backslash character (\) 23
- .bat extension 9

- batch files 35–36, 39
 - autosend directory 9
 - creating fax envelopes 7
 - delivery of 36
 - email-to-fax method 11
 - structure of 35
 - vs. tag files 18, 35
- blind sends 9
- blind submittal 71

C

- C++ projects 14
- client computers 25
- client workstations 25
- COM (Component Object Model) classes 14
- command-reference 60
- commands
 - @+IMAGE 21–24
 - embedding 21–24
 - lp 16, 50, 51
 - lp spooler 49–51
 - spool 53
 - vfx 7–8, 67–71
 - vfxadmin config 29
 - vinfo filetypes 70
 - xmlf 13, 58
- common section 35
- Component Object Model (COM) classes 14
- configuring
 - email-to-fax feature 27–29
 - WordPerfect integration 53–54

- content element 60
- conventions
 - typeface 5–6
- cover pages
 - adding 70
 - described 2
 - faxing cover page only 71
 - listing 71
 - selecting 55
- custom printer interface scripts 16, 50

D

- database programs 31
- databases 36
- DDE (Dynamic Data Exchange) print entry 70
- delimited text files 37
- delimiter statements 36
- destination data 32, 33
- devices
 - See also* stdin; stdout; modems
- directories
 - autosend directory 9–10
 - for Linux interfaces 49
 - for WordPerfect installation 54
- drivers. *See* print drivers
- DTDs (Document Type Definitions) 73–83
- Dynamic Data Exchange (DDE) print entry 70

E

- electronic business forms 43–44
- email
 - autofax account 27–30
 - creating fax envelopes 11
 - email address in subject line 29
 - email-to-fax feature and 27
 - fax number in subject line 29
 - message formats for 29
- email accounts 11, 28
- email addresses 2, 69
- email-to-fax feature 11, 27–30
- embedded items
 - fax numbers 33
 - files 18
 - tags 8
 - vfx commands 21–24
- End of File (EOF) character strings 19

- enterprise-wide faxing 8
- envelopes. *See* fax envelopes
- environment variables 6
- EOF (End of File) character strings 19
- errors
 - autosend directory and 9
 - “Permission Denied” error 70
- Extensible Markup Language (XML) 3, 8, 13, 57
- extensions 9

F

- “fast-save” feature 55
- fax automation classes 14
- “fax blasting” 31
- Fax Cancel DTD 80
- Fax Cancel Response DTD 81
- fax clients 25
- fax envelopes 2–4
 - contents of 2
 - creating 11
 - creation order 36
 - delivery mechanisms for 5–6
 - print-to-fax feature 25–26
 - sending to fax server 67
- fax merge feature 32–35
 - embedded tags and 23
 - fax numbers and 35
 - file structure 33
 - mail merge feature and 31
 - template files and 37–38
- fax numbers
 - embedded numbers 33
 - fax merge and 35
 - including in email subject line 29
 - tfn tag and 3
- fax request ID. *See* request ID
- fax requests
 - canceling 65
 - ID numbers for 68
 - status of 64
- Fax Resource DTD 82
- Fax Resource Response DTD 83
- fax servers
 - autosend directory and 9
 - delivering faxes to 4
 - email-to-fax feature 27–30

- fax merge and 35
 - identifying group files to 37
 - networked to mail server 11
 - passing delimited text files to 36
 - passing tags to 17
 - print-to-fax feature 25–26
 - sending fax envelopes to 67
 - virtual fax server 47–48
 - XML-F and 57
 - Fax Status DTD 77
 - Fax Status Response DTD 78–79
 - Fax Submit DTD 9, 73–75
 - Fax Submit Response DTD 76
 - fax-cancel document 65
 - faxes
 - See also* fax envelopes
 - adding logos to 21
 - adding signatures to 21
 - attachments 67
 - creating fax envelopes 3
 - creating printer destination for 49
 - delivering fax envelopes to fax server 4
 - group faxes 18
 - integration services for 5–16
 - merging 32–35, 37–38, 42
 - overlying forms onto 39–45
 - removing headers from 43
 - scanning 43, 44
 - sending with vfx command 67–71
 - faxing
 - email-to-fax feature 11, 27–30
 - enterprise-wide faxing 8
 - from WordPerfect 54
 - group faxing 31–38
 - lp faxing 16
 - print-to-fax feature 12, 25–26
 - fax-status document 63
 - fax-status-response document 64
 - fax-submit document 59–62
 - fax-submit-response document 62
 - faxwpc.sh file 54
 - faxwpdoc.sh script 55
 - faxwpx.sh file 53
 - file extensions 9
 - files
 - See also individual files*
 - batch files 9, 11, 18, 35–36
 - embedded files 18
 - extensions for 9
 - fax-cancel document 65
 - fax-status document 63
 - fax-status-response document 64
 - fax-submit document 59–62
 - fax-submit-response document 62
 - group files 31, 36
 - “here” documents 18
 - local files 18
 - logo files 44–45
 - merging fax files 42
 - Microsoft Word files 33
 - PCL files 8, 34
 - PostScript files 8
 - private files 44
 - public files 44
 - send documents 60–62
 - signature files 44–45
 - tag files 7, 9, 11, 18
 - template files 31
 - text files 21–24
 - TIFF files 21
 - word processing files 33
 - XML documents 13
 - XML-F documents 8, 9, 13
 - fil tag 3, 18, 19
 - fnm tag 12, 36, 38
 - forms overlay 39–45
- ## G
- Graphical User Interfaces (GUIs) 14
 - group faxing 18, 31–38
 - group files 31, 36
 - GUIs (Graphical User Interfaces) 14
- ## H
- “here” documents 18
 - hplj.prs file 54
 - hyphen (-) 41
- ## I
- IBM AIX systems 49
 - images

- shrinkage of 43
- specifying position of 22–24
- InstallPrinter program 14
- integration techniques, fax services 5–16

L

- letterhead, overlaying on fax 39
- lib/images directory 22
- Linux platform
 - embedded commands and 24
 - format notations for 6
 - interfaces directory 49
 - lp spooler 16
 - printer interface scripts 49
 - SDK and 14
 - WordPerfect macros and 15
- local files 18
- logos
 - creating 44–45
- logos, adding to faxed documents 21
- lp command 16, 50, 51
- lp faxing 16
- lp spooler 16
- lp spooler command 49–51

M

- macros, WordPerfect 15
- mail merge feature 31, 32
- mail servers 11
- manuals, obtaining additional 7
- merge option 42
- merging faxes 32–35, 37–38
- merging files 42
- Microsoft Visual Basic 6.0 14
- Microsoft Visual C++ 6.0 14
- Microsoft Word files 33
- MMC snap-in 28

N

- notes, explanation of 6
- ntf tag 18

O

- Outlook print driver 15
- overlays

- multiple overlays 41
- single overlay 40

P

- passwords 29
- path statements 6
- PCL files 8, 34
- PCL print driver 12, 25, 26
- PCL5e code 43
- “Permission Denied” error 70
- pipe character (|) 36
- pipe-delimited ASCII format 36
- POP3 mail gateway 11
- POP3 server 27–29
- PostScript code 43
- PostScript files 8
- PostScript print driver 12, 25, 26
- print drivers
 - conventional 26
 - described 34
 - fax 16
 - Outlook 15
 - VSI 12, 14, 25, 34, 54
 - WordPerfect 53
- print spooler 55
- Print to File command 34
- “Printer action required” message 55
- printer interface scripts 49, 50, 51
- printers
 - ASCII printers 49
 - creating 54
 - creating printer destination 53
 - fax merge jobs and 34
- printing
 - creating fax printer destination 49
 - print-to-fax feature 12, 25–26
- print-to-fax feature 12, 25–26
- private files 44
- problems. *See* troubleshooting
- public files 44

R

- recipient element 59
- recipient information 2, 36, 59
- recipient list 32
- recipient sections 35

request ID 68
 request-results element 62
 rned 62

S

scanning faxes 43, 44
 scripts
 ASCII text printer interface scripts 49
 custom printer interface scripts 16, 50
 lp faxing 51
 printer interface scripts 49, 50, 51
 sample interface script 51
 SDK (Software Development Kit) 14
 send documents 60–62
 send parameters 2
 sender element 59
 sending faxes 2–4, 67–71
 servers
 See also fax servers
 mail servers 11
 POP3 server 27–29
 Windows fax servers 70
 service-reference element 62
 shell environment 70
 signatures
 creating 44–45
 signatures, adding to faxed documents 21
 Software Development Kit (SDK) 14
 spool command 53
 spooler 55
 spreadsheet programs 31
 standard input device. *See* stdin
 stdin (standard input device)
 vtiff tool utility and 41
 sub tag 38
 subject element 59
 system requirements 48

T

.tag extension 9
 tag files
 autosend directory 9
 creating fax envelopes 7
 described 18
 email-to-fax method 11
 specifying overlay forms on 39

vs. batch files 35

tags

See also individual tags
 creating fax envelopes 7
 embedded tags 8
 passing to fax server 17
 print-to-fax method 12
 tag order 36
 tag processing 37
 vfx tags 17–19

technotes 7

template files 12, 31

text files

delimited text files 37
 embedding vfx commands in 21–24
 passing to fax server 36

text overlay 40

tfn tag 3, 17, 36

tgf tag 17

TIFF files 21

electronic business forms and 43
 forms overlay and 39
 scanning faxes into 44

tilde character (~) 23

tips, explanation of 7

typeface conventions 5–6

U

Universal Resource Locator (URL) 58

Unix platform

embedded commands and 24
 format notations for 6
 interfaces directory 49
 lp command 50, 51
 lp faxing 16
 lp spooler 16
 printer interface scripts 49
 SDK 14
 WordPerfect integration 53
 WordPerfect macros 15

URL (Universal Resource Locator) 58

V

vfx command 7–8, 67–71
 description 67–68
 embedding in ASCII files 21–24

- options for 68–71
- sending faxes with 67–71
- synopsis 67
- vfx tags 3, 17–19
- vfxadmin config command 29
- vinfo filetypes command 70
- virtual fax server 7, 47–48
- Visual Basic projects 14
- VPOP section 28
- “vsi” macro 55
- VSI MMC Fax Server Administration snap-in 28
- VSI print drivers 12, 14, 25, 34, 54
- VSI Software Development Kit (SDK) 14
- vsi.wpm file 54
- VSI-FAX
 - choosing integration strategy 6
 - fundamentals 1–16
 - obtaining additional manuals for 7
 - platforms not supported 13
 - WordPerfect integrations 53–55
- vsifax.ini file 71
- vsipcl.prs file 54
- vsipclx.prs driver 54
- vsipclx.prs file 53
- vsisrv.ini file 28, 29
- vsiwp.env file 54
- vtifftool clear command 43
- vtifftool merge command 42
- vtifftool overlay command 41

W

- warnings, explanation of 6
- web fax client print driver 15
- Windows fax servers 70
- Windows platform
 - applications for 14
 - embedded commands and 24
 - format notations for 6
 - lp spooler and 16
 - sending faxes from Windows applications 14
- word processor programs 31, 32–34
- WordPerfect
 - configuring integration 53–54
 - faxing documents from 54
 - installation directory 54
 - macros 15

- non-graphical integration 54–55
- WordPerfect for Unix macros 17
- WordPerfect integrations 53–55
- WordPerfect print driver 53
- wpc.config file 54
- wpc.config script 54
- wprint7c utility 55
- wprintc utility 55

X

- X Windows 53–54
- XML (Extensible Markup Language) 3, 8, 13, 57
- XML documents 13
- .xml extension 9
- XML-F 57–65
- xmlf commands 13, 58
- XML-F documents 8, 9
- XML-F DTDs 73–83
- XML-F Fax Submit DTD 9, 13
- XML-F service 64
- XML-F specification 3
- xwpdest program 53