



Version 9.4
Integration Module Guide

v. 1.0
September 17, 2008

Edition

Information in this document applies to version 9.4 of the RightFax Integration Module Guide.

Copyright Notice

©2008 All rights reserved. No part of this publication may be reproduced, transmitted, transcribed, stored in a retrieval system, or translated into any language in any form by any means without the written permission of Captaris. The following are registered trademarks and trademarks of Captaris Inc. and its subsidiaries: Captaris, the Captaris logo, Alchemy®, Captaris Workflow™, RightDocs™, RightFax®, RightFlow™ and RightStar™ in the US and/or other jurisdictions. FaxPress™ is a registered trademark of Castelle. RecoStar, DOKuStar, DOKuStar Capture Suite, Single Click Entry, and Invoice CENTER are registered trademarks and trademarks of Captaris Document Technologies GmbH. All other brand names and trademarks are the property of their respective owners.

For More Information

Captaris, Inc. is a leading provider of software products that automate document-centric business processes. Captaris specializes in document capture, recognition, routing, workflow and delivery. Captaris integrated solutions provide interoperability with leading line of business applications and technology platforms. Captaris products include RightFax, Captaris Workflow, Alchemy, FaxPress, DOKuStar, RecoStar, Single Click Entry and IDStar which are distributed through a global network of leading technology partners. Captaris customers include the entire Fortune 100 and the majority of Global 2000 companies. Headquartered in Bellevue, Washington, Captaris was founded in 1982 and is publicly traded on the NASDAQ Global Market under the symbol CAPA. www.Captaris.com.

Contents

Chapter 1	Introduction..... 7	Chapter 5	Using the Integration Setup Wizard.....21
	Using This Guide.....7	Chapter 6	Configuring the Integration Module to Receive Data25
	Other Resources for RightFax Users8		Creating an Input Device25
	Getting Help Online8		Setting Up a Named Pipe Capture.....27
	Captaris Training Services.....8		Setting Up Directory Scanning.....28
	Customer Support9		Setting Up a Serial Capture29
Chapter 2	Overview 11		Setting Up a TCP/IP Port Capture31
	Features of the Integration Module..... 11		Setting Up an IBM WebSphere MQ Connection.....32
	Understanding Document Recognition 12	Chapter 7	Testing a Connection and an Input Device35
	Understanding Document Distribution..... 12		Testing a Named Pipe Connection35
Chapter 3	Installing the RightFax Integration Module 15		Testing Directory Scanning35
	Installing the Integration and Business Integration Modules 15		Testing a Serial Connection 36
Chapter 4	Connecting to the RightFax Server 17		Testing a TCP/IP Port Connection 36
	Guidelines for Common Connection Methods 18		Testing an IBM WebSphere MQ Connection..... 36
		Chapter 8	Creating FCL Documents.....39
			Understanding the Format of FCL Commands.....41
			Example FCL Documents41

Chapter 9	Setting Defaults for FCL Documents 45	Chapter 14	Creating and Attaching Cover Sheets 77
	Opening the Integration Module Configuration Program45		Understanding Production Cover Sheets 78
	Setting General Defaults45		Creating Cover Sheet Templates 78
	Setting Defaults for FCL Processing 47		Attaching a Cover Sheet to a Document with FCL.....82
Chapter 10	Creating Filter Templates 49		RightFax Cover Sheet Logic 83
	Understanding Filter Templates..... 49		Example Cover Sheet and Document 84
	Preparing to Create Filter Templates51	Chapter 15	Creating Lines, Boxes, and Other Shapes 87
	Creating a New Filter Template51		Understanding the Commands That Create Shapes.....87
	Understanding the Field Types 59		Using the Most Common Commands That Create Shapes.88
	Troubleshooting.....63	Chapter 16	Setting Margins and Tabs 93
	Creating an Input Channel for Filter Documents..... 64		Setting Margins.....93
Chapter 11	Specifying the Destination of the Document 65		Starting a New Page With the Same Margins as the Previous Page94
	Sending One Document to a Single Fax Number.....65		Setting Tabs with FCL94
	Sending One Document to Many Recipients 66	Chapter 17	Selecting and Configuring Fonts 97
	Sending One Document to Many Recipients With an Include File..... 66		Understanding TrueType Fonts..... 97
	Creating Lists of Recipients for a Broadcast.....67		Understanding Support for Other Fonts 97
	Using Cover Sheets in a Broadcast..... 69		Selecting Fonts98
	Sending Documents to a Printer or File with FCL 70		Changing the Appearance of Fonts98
Chapter 12	Creating and Linking Background Forms 71	Chapter 18	Setting Page Orientation and Image Quality 101
	Understanding Background Forms 71		Setting Page Orientation 101
	Creating Background Forms.....71		Setting Fax Image Quality..... 102
	Linking Background Forms..... 72	Chapter 19	Attaching and Embedding Files, Signatures, and Graphics 105
Chapter 13	Including the Same Files and Commands in Many Documents 73		Attaching Files to Documents with FCL 105
	Understanding Include Files..... 73		Embedding Signatures with FCL 106
	Creating Include Files 74		Embedding Graphics with FCL..... 108
	Linking Include Files.....75		

Chapter 20	Scheduling Document Transmission	109	Chapter 25	Programming for the RightFax API for Java.....	165
Chapter 21	Setting Up Notification Messages of Document Transmission	111		Installing the API for Java.....	166
	Creating Notification Messages with FCL.....	111		Sending Outbound Documents	167
	Creating Notification Templates	112		Querying Documents.....	170
	Creating Notification Channels	118		Performing Actions on Documents.....	172
	Including the {{Notifyhost}} Command in Documents.....	128		Using Debug Mode.....	174
Chapter 22	Testing and Troubleshooting Notification Messages.....	131	Chapter 26	Creating FCL Documents with InternetLink Commands 177	
	Testing That the Host Application Is Correctly Receiving Notifications	131		System Requirements.....	177
	Troubleshooting.....	132		Activating the InternetLink Module	177
Chapter 23	Setting Up Actions on Document Transmission	135		Understanding the InternetLink FCL Commands	178
	Printing or Faxing a Copy of a Document That Was Transmitted.....	135		Attaching Native Files to InternetLink Documents.....	179
	Printing or Faxing a Document That Cannot Be Transmitted 137			Sending Documents as E-mail if Faxes Fail.....	181
	Performing Actions on Documents With Missing Data	138		Receiving Notification When a Fax Fails and Is Sent as an E-mail.....	181
Chapter 24	Programming for the RightFax XML Interface.....	141		Standard FCL Commands with InternetLink Options.....	181
	Introduction to the RightFax XML Interface.....	141		InternetLink FCL Commands	182
	Installing the XML Interface.....	141	Appendix A	Integration Module Programs	187
	XML Interface Functions	142	Appendix B	FCL Commands	189
	Transports.....	143		Index	221
	Understanding Body and Cover Text	147			
	The Schemas	147			
	Sample Documents Based on the Schemas	157			

Chapter 1

Introduction

The RightFax Integration Module enables applications for information exchange by integrating with applications on mainframe, mid-range, and local area network host systems. Together, the RightFax server and the Integration Module will send any document created by these applications via fax, e-mail, or over the Internet.

The RightFax Integration Module automates batch-oriented, repetitive processes. It is designed to support applications that produce output that traditionally is sent to a printer, printed on pre-printed forms, folded, stuffed in envelopes, and then mailed or manually faxed. These documents can include invoices, itineraries, purchase orders, statements, order confirmations, loan applications, bills of lading, change orders, financial reports, and material safety data sheets.

The RightFax system can integrate with many applications. The integration options are listed in the following table, along with the RightFax documentation where you can refer for more information.

Table 1a RightFax Integration Options

Integration option	Refer to these resources for information
Integration Module services on the RightFax server with Facsimile Command Language (FCL)	Chapters 2-24 in this guide <i>RightFax Administrator's Guide</i>
RightFax XML Interface	Chapter 24, "Programming for the RightFax XML Interface" in this guide
RightFax API for Java	Chapter 25, "Programming for the RightFax API for Java" in this guide
RightFax C, C++, and Visual Basic API	www.captaris.com/

Using This Guide

The *RightFax Integration Module Guide* contains an overview for configuring and working with the RightFax Integration Module. It is written for RightFax administrators. It is a reference guide, with

introductions to and overviews of technical topics. This guide is intended to be used with the *RightFax Administrator's Guide*, and it is supplemental to Captaris technical training classes.

Document conventions

Notes and warnings in this guide provide instructions for special circumstances, side effects and product interactions, and important reminders.

- Notes include information that you may find useful but do not affect the integrity of your computer hardware, software, or data.
- Warnings always indicate that failure to proceed carefully may result in loss of data or damage to hardware or software. Always read and understand warnings before proceeding.

Because most RightFax products let you customize the install location, paths are not presented as literal paths from the root folder. Instead, the install folder you selected at the time of installation is assumed to precede folder names where applicable.

If the text in this guide contains user input (such as command-line commands with switches) that should be typed exactly as shown, the user input is **bold**. If the user input contains variables (such as variable parameters in a command line), the variables are *italicized* and then described in the subsequent paragraph.

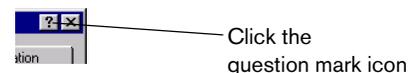
Other Resources for RightFax Users

Because your RightFax servers may support dozens, hundreds, or even thousands of fax users, RightFax has designed its end-user documentation to be thorough, easy-to-use, and easily accessible.

In addition to the user's guides, Quick Reference Cards give you quick and easy access to the most common fax management features of the FaxUtil, Microsoft Outlook, and Lotus Notes. These useful cards are compact and can be easily distributed throughout even the largest organizations.

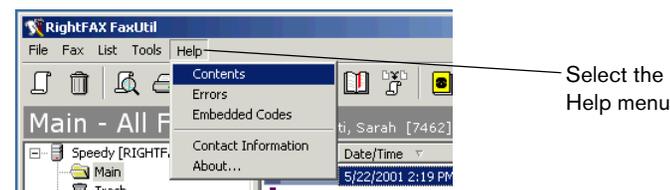
Getting Help Online

Most of the RightFax windows and dialog boxes include context-sensitive help. Typically, this help describes each field and option in a dialog box. This context-sensitive help can be launched from the question mark icon in the title bar of a dialog box.



- Click this icon, and then click any option on the dialog box to view its definition.
- Or, press F1 to view a definition.

Online help for tasks or conceptual topics also is available. View this help by selecting the **Help** menu. You can browse the help for a topic or use the index to look up key words and terms.



The Captaris Web site offers the latest product information. Updated documentation, a searchable customer support knowledgebase, software downloads, and the latest product offerings are all online at www.captaris.com/.

Captaris Training Services

RightFax training gives you the skills to optimize your RightFax solution. Our comprehensive technical, administrative, and user training programs produce full utilization and understanding of RightFax products. Regular classes are held, throughout the year.

Training materials and computer-based training tools are also available. For more information on RightFax training, please visit our Web site at www.captaris.com/training and let us help you develop a customized training plan for your organization.

Customer Support

Your *Customer Support Guide* includes detailed information about the support options available to RightFax customers. Please fill out the *RightFax Software Warranty & Registration Card* and return it immediately. If you have questions of a technical nature, contact your organization's RightFax administrator or network administrator before calling the RightFax customer support department. On the RightFax Web site, a database of customer support knowledge contains a wealth of information on installing, configuring, and maintaining RightFax software.

RightFax Customer Support:

Captaris	Voice: (520) 320-7070
155 N Rosemont Blvd Suite 101	Web:
Tucson, Arizona 85711 USA	www.captaris.com/support/index.html
	E-mail: support@captaris.com

5:00 A.M. to 5:00 P.M. Pacific time, Monday through Friday

7:00 A.M. to 3:00 P.M. Pacific time, Saturday

■ ■ ■

Chapter 2

Overview

This chapter provides an overview of the Integration Module services that can be implemented on the RightFax server with Facsimile Command Language (FCL). It describes the features of the Integration Module and the workflow and programs that recognize data, assemble the data into documents, distribute documents, and report the results.

Features of the Integration Module

The Integration Module provides the following features.

Formatting documents

The Integration Module can:

- Add lines, boxes, and other shapes to a document.
- Set fonts, margins, and tabs.
- Add graphics to a document, such as a signature or a company logo.
- Add a background form over which the document data is placed, such as a purchase order form, bill of lading, or itinerary.

Scheduling documents

In addition to the scheduling control provided by the RightFax server, the Integration Module can:

- Delay the sending of a document by minutes or schedule the date and time for a document to be sent.
- Prioritize a document for sending.
- Hold a document for approval.
- Send documents in batches.
- Broadcast faxes.

Attaching cover sheets and other attachments

When a document is processed by the Integration Module, additional documents can be attached, including cover sheets. Information for the cover sheet can be provided by FCL codes.

Including data from a lookup table

A document from a host-based application may not include all the information that is needed to send it, such as a fax number. You can create a table of data where the Integration Module can “look up” the needed data.

Sending notification messages

As a document is sent via the RightFax server, notification messages can be generated and sent back to the sender of the document, to a system administrator, to a central mailbox, to a file, to a directory, or to a database on the host application. They can be e-mailed or faxed, or files can be transferred.

Notification messages can be customized to contain descriptive information about the sent document, such as:

- Sender and recipient information, such as name, company name, fax number, voice number, and e-mail address.
- Document data such as the number of pages, transmission date and time, image quality, and the duration of the fax call.
- Status of the fax from the fax board.

Another form of notification is to fax or print documents that are sent or documents that encounter errors in sending.

Understanding Document Recognition

The Integration Module can process ASCII text files, PostScript files, and print control language (PCL) files that are generated by an application.

ASCII text files

To process text files, you must do one of the following:

- Include FCL commands in the document data. This may require custom programming to add FCL to documents or to templates in the host application or to insert FCL in the data stream. This is known as *native mode*.
- Create “filter templates” that add FCL to the document data after it is sent to the Integration Module for processing. A filter template is a map of the document data that contains the FCL that is required to create and send the document. This is known as *filter mode*.

PostScript files

To process and transmit a PostScript file, the host application must send a “false” first page with the PostScript document. This page must contain FCL codes in text form, which can be interpreted by the Integration Module. The Integration Module interprets the FCL, removes the false first page, formats the document, and transmits it. This may require custom programming.

PCL files

To process and transmit a PCL file, the host application must send a “false” first page with the PCL document. This page must contain FCL codes in text form, which can be interpreted by the Integration Module. The Integration Module interprets the FCL, removes the false first page, formats the document, and transmits it. This may require custom programming.

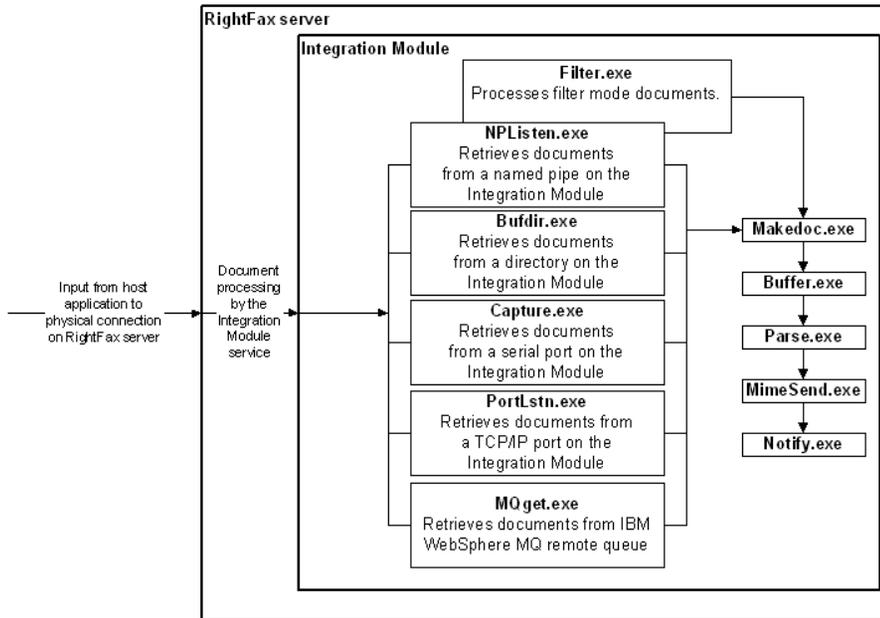
Understanding Document Distribution

The following figure shows how the Integration Module programs receive, process, recognize, and distribute data from the host application. This illustration shows the input channels that can be configured to receive and recognize data. Up to 70 channels can be configured.

Example The executable Bufdir.exe scans a folder for files sent by the host application. Bufdir.exe retrieves the files and sends them to Makedoc.exe, which begins the process of converting them to documents and transmitting them. Notify.exe creates a notification that is sent to an application on the host system.

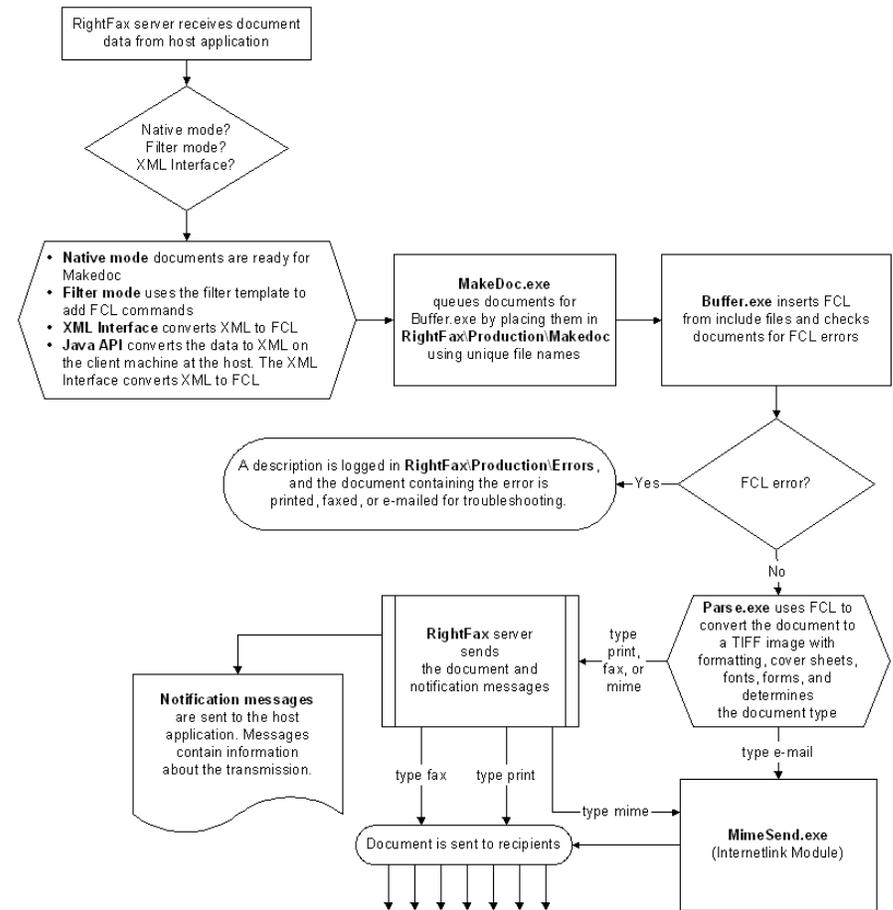
For a list of the programs that process and send documents, see [Appendix A, "Integration Module Programs"](#).

Figure 2.1 Flow of Documents in the Integration Module



"Figure 2.2: Integration Module Data Flow" shows how FCL documents are processed after the Integration Module receives the data stream from a host application. It explains the programs, such as Makedoc.exe, that are shown briefly in "Figure 2.1: Flow of Documents in the Integration Module".

Figure 2.2 Integration Module Data Flow



Chapter 3

Installing the RightFax Integration Module

The RightFax Integration Module enables applications for information exchange by integrating with applications on mainframe, mid-range, and local area network host systems. To meet the needs of small, medium, and large enterprises, the RightFax Integration Module comes in two versions: the Integration Module which includes full functionality, and the Business Integration Module which includes a limited set of features for use by smaller organizations.

Table 3a RightFax Integration Module types

Module	Features
RightFax Integration Module	<ul style="list-style-type: none">• 70 input channels.• Output methods of fax, certified email, encrypted email, or print.• Up to 70 filter templates.• Up to 128 notification channels.• Unlimited notification messages.
RightFax Business Integration Module	<ul style="list-style-type: none">• Two input channels.• One output method (fax or print) per input channel.• Two fax channels for each fax output.• One filter template per input channel.• Unlimited notification messages.

Installing the Integration and Business Integration Modules

The RightFax Integration Module and Business Integration Module software are installed automatically during the RightFax server installation. To enable the functionality of these modules, they must be licensed and then simply enabled on the RightFax server. For information on enabling or adding new components to a RightFax server, refer to the *RightFax Installation Guide*.

To install the RightFax Enterprise Integration Module

1. Install the RightFax server, as described in the *RightFax Installation Guide*.
2. Configure the RightFax server, as described in the *RightFax Administrator's Guide*.
3. If the Integration Module service does not appear in Enterprise Fax Manager, you need to create the service. To do so, open Windows Control Panel. Double-click the **RightFax Integration** applet. The **Integration Module Configuration** opens. Click **OK**, and the service will be created.

4. Configure the Integration Module software, as described later in this guide.

■ ■ ■

Chapter 4

Connecting to the RightFax Server

This chapter explains the most common methods for connecting a host computer to the Integration Module computer for FCL-based documents.

Because several methods can be used to connect a host computer to the Integration Module on the RightFax server, you must determine the best choice for the capabilities of the host system environment and the distribution methods for documents and notifications about sent documents.

Because of the wide range of host applications, environments, and potential connection scenarios, this chapter does not provide step-by-step connection procedures.

The following figures illustrate the most common connection methods. Some connections require third-party software that

Captaris does not provide or support.

Figure 4.1 Connections to Applications that Generate FCL or ASCII Data

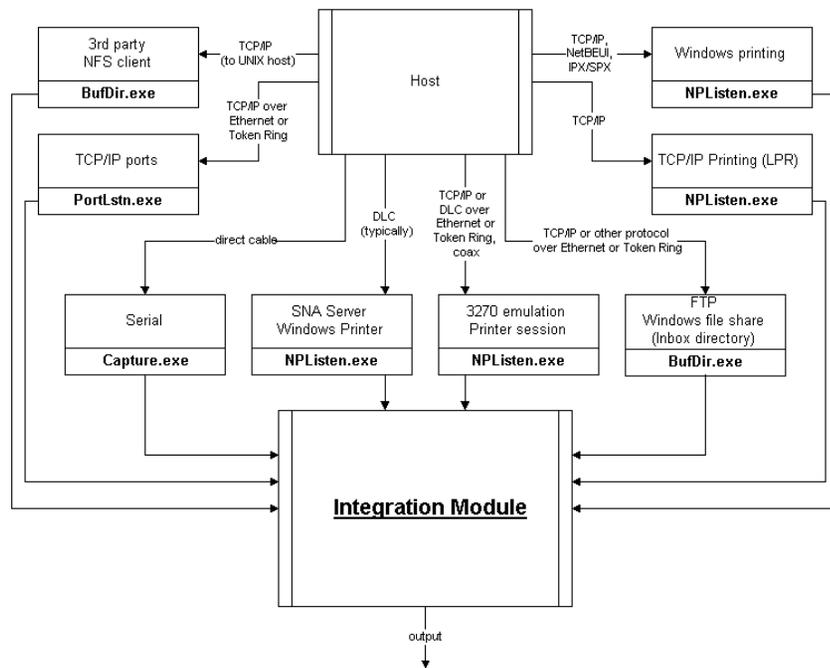
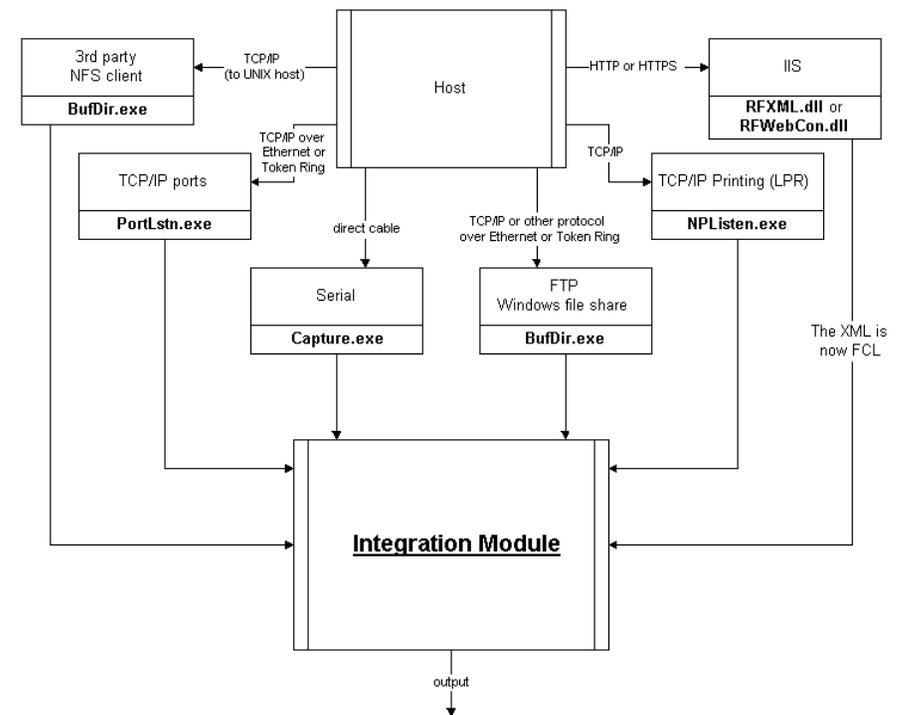


Figure 4.2 Connections to Host Applications that Generate XML and Java Data



Guidelines for Common Connection Methods

The most common communication connection methods and basic requirements are described in the following sections:

- Line printer remote (LPR) connection on [page 19](#)
- 3270 emulation on [page 19](#)
- TCP/IP socket connection on [page 19](#)
- File Transfer Protocol (FTP) connection on [page 19](#)
- 5250 emulation on [page 19](#)
- IBM® WebSphere MQ® client v6 for Microsoft Windows on [page 20](#)

After the communication methods are established, you must configure the Integration Module to receive data via those methods, as described in [Chapter 6, “Configuring the Integration Module to Receive Data”](#).

In general, the RightFax Integration Module is a printer connected to the host computer. The host application will print to the Integration Module.

Line printer remote connection

To use a line printer remote connection, set up the host system to print to the RightFax Integration Module as a remote Berkeley Style Device (BSD) printer. In most cases, set the remote host name to the host name of the Integration Module, and set the remote printer name to “hostfax”. (By default, the Integration Module installs a printer called “hostfax”. This is usually the default printer.)

The TCP/IP printing service or Print Services for Unix must be installed and started on the RightFax server.

To configure the Integration Module to receive data via this method, see [“Setting Up a Named Pipe Capture”](#) on [page 27](#).

3270 emulation

You can set up a direct connection from the RightFax server to a mainframe host with 3270 emulation software. The protocol standard TN3270 using TCP/IP over an Ethernet network adapter is the recommended method. Systems Network Architecture (SNA) can be used. Coaxial cable with a 3270 coaxial adapter can be used, but it is not recommended.

Emulation software often has difficulty with a coaxial cable connection in a Windows environment. TN3270 is often a better choice and is easier to configure if the mainframe has a TCP/IP connection.

In the event that connections are required from different regions or different host computers, multiple sessions are supported up to the limits of the adapter hardware and software. For example, a 3270 coaxial adapter typically will support five separate printer or terminal sessions to a single cluster controller.

To send notification messages back to an application on the host system, a separate connection method (known as a connection channel) can be defined.

To configure the Integration Module to receive data via this method, see [“Setting Up a Named Pipe Capture”](#) on [page 27](#).

TCP/IP socket connection

You can set up a direct socket connection via TCP/IP to and from the RightFax server through any available port. This usually requires that you acquire or create connection software for the host system.

To configure the Integration Module to receive data via this method, see [“Setting Up a TCP/IP Port Capture”](#) on [page 31](#).

File Transfer Protocol connection

The RightFax Integration Module can use FTP server services to accept documents from the host computer. The Integration Module also can use an FTP client to send notification messages back to an FTP server on the host system.

For FTP connections, you must install and configure an FTP server on the RightFax server computer.

To configure the Integration Module to receive data via this method, see [“Setting Up Directory Scanning”](#) on [page 28](#).

5250 emulation

To send documents from the host computer to the Integration Module, 5250 emulation software is recommended. A serial connection with a protocol converter can be used, but it is not recommended.

The protocol converter sends data to the RightFax server via a serial cable connected between the converter and one of the COM ports on the RightFax server. If this method is used, the Integration Module must be configured to accept input on the serial port.

To send notification messages back to the host, SQL via ODBC is recommended. Emulation software via 32-bit high level language application programming interface (HLLAPI) or via a serial connection also can be used.

To configure the Integration Module to receive data via this method, see [“Setting Up a Serial Capture”](#) on [page 29](#).

IBM WebSphere MQ client for Windows

The RightFax Integration Module can communicate with an IBM WebSphere MQ 6 channel via TCP/IP. To configure the Integration Module to receive data via this method, see [“Setting Up an IBM WebSphere MQ Connection”](#) on [page 32](#).

■ ■ ■

Chapter 5

Using the Integration Setup Wizard

The RightFax Integration Module includes a wizard that helps you to create configurations for many of the connection methods. Use the wizard to:

- Configure the Integration Module service and set defaults for documents that are sent from the Integration Module.
- Configure input channels from the host application to the Integration Module.
- Set notification actions and messages. You can choose to print or fax copies of documents as they are transmitted. You also can define messages with descriptive information about the documents as they are transmitted.
- Format documents with filter templates.

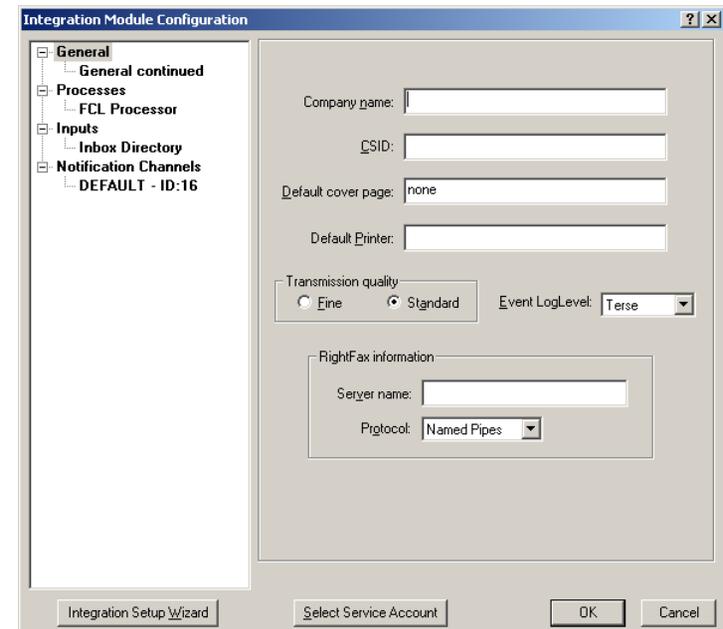
The Integration setup wizard is designed to guide you through each step of configuring the Integration Module. Instructions in this guide supplement the instructions in the wizard. Captaris recommends that you review the topics in this guide before you begin using the wizard.

To start the wizard

1. On the **Start** menu, select **Programs > RightFax > Enterprise Fax Manager**. The **Enterprise Fax Manager** window opens.
2. In the **Fax Servers** list, click the name of the server where the Integration Module is running.

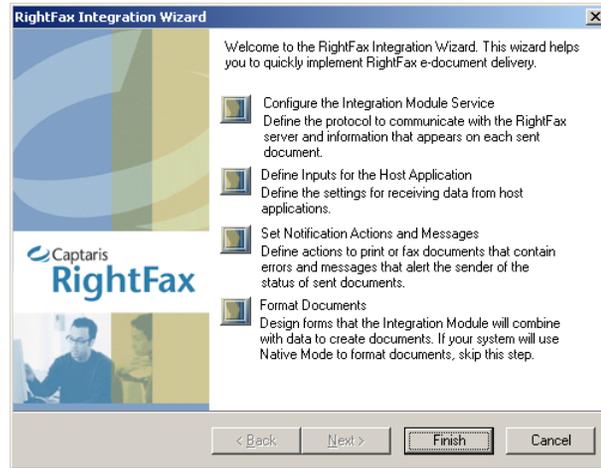
3. In the **Service Name** list, double-click **RightFax Integration Module**. The **Integration Module Configuration** window opens.

Figure 5.1 The Integration Module Configuration Window



4. Click **Integration Setup Wizard**. The wizard starts, and the **Integration Setup Wizard** window appears.

Figure 5.2 The Integration Setup Wizard



This window is the starting point for the configuration. Select from the configuration tasks, as described in the following sections.

Configure the Integration Module service

In this series of steps, you can define information that will appear on each fax that is sent from the Integration Module. Some of these settings can be overridden with facsimile command language (FCL) in the documents that are sent.

You can make the following default settings for documents:

- Name and fax number to appear on the cover sheet.
- File to send as the cover sheet.
- The default printer for documents sent from the Integration Module.
- Fax image quality.
- Page length, page size, and reducing the image size to fit the page.

These settings are also described in [Chapter 9, “Setting Defaults for FCL Documents”](#).

Define inputs for the host application

In this series of steps, you can define the settings for receiving data from host applications via a named pipe, directory, TCP/IP, IBM WebSphere, or a custom input type.

For each communication method, you will specify the communications protocol and transmission method, port, directory, or queue to monitor, and the configurations needed for each protocol. You will:

1. Name the input.
2. Specify the input type.
3. Configure the input.

These settings are also described in [Chapter 6, “Configuring the Integration Module to Receive Data”](#).

Set notification actions and messages

In this series of steps, you can define notification actions and messages.

Notification actions can:

- Fax sent documents, whether successfully or not successfully transmitted, to an internal fax number.
- Print or fax documents that cannot be successfully transmitted because data is missing.

Notification messages can:

- Notify users that a document was sent.
- Notify an administrator of the status of documents.

Notification messages can provide descriptive information, such as whether or not a document was transmitted, explanations of errors, and transmission duration, dates, and times. These messages are sent to a host application from the Integration Module when it processes and sends documents.

These settings are also described in:

- [Chapter 21, “Setting Up Notification Messages of Document Transmission”](#)
- [Chapter 23, “Setting Up Actions on Document Transmission”](#)

Format documents

In this series of steps, you can create filter templates for documents with MapText. These settings are also described in [Chapter 10, “Creating Filter Templates”](#).

■ ■ ■

Chapter 6

Configuring the Integration Module to Receive Data

This chapter describes the procedures for configuring the RightFax Integration Module to receive data. You will create an input device for receiving or retrieving data by writing a command line for each type of input needed in the system.

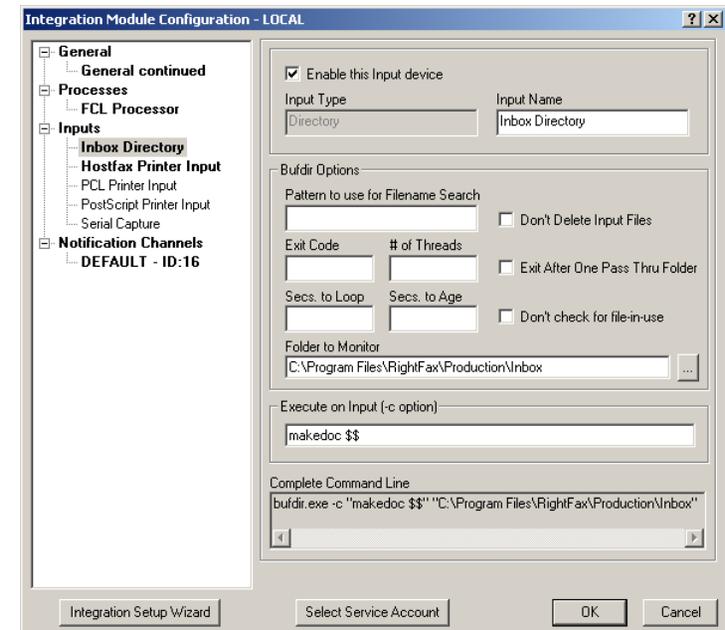
Note that command lines are case-sensitive.

Creating an Input Device

1. On the **Start** menu, select **Programs > RightFax > Enterprise Fax Manager**. The **Enterprise Fax Manager** window opens.
2. In the **Fax Servers** list, click the name of the server where the Integration Module is running.
3. In the **Service Name** list, double-click **RightFax Integration Module**. The **Integration Module Configuration** window opens.

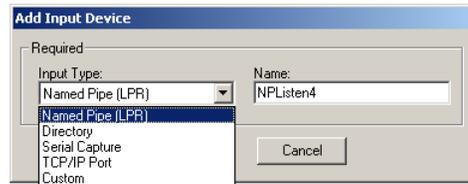
4. In the left pane of the **Integration Module Configuration** window, click **Inputs**. The input settings appear.

Figure 6.1 The Inputs Settings



- Right-click **Inputs**, and select **Add Input Device** from the shortcut menu. The **Add Input Device** dialog box appears.

Figure 6.2 The Add Input Device Dialog Box



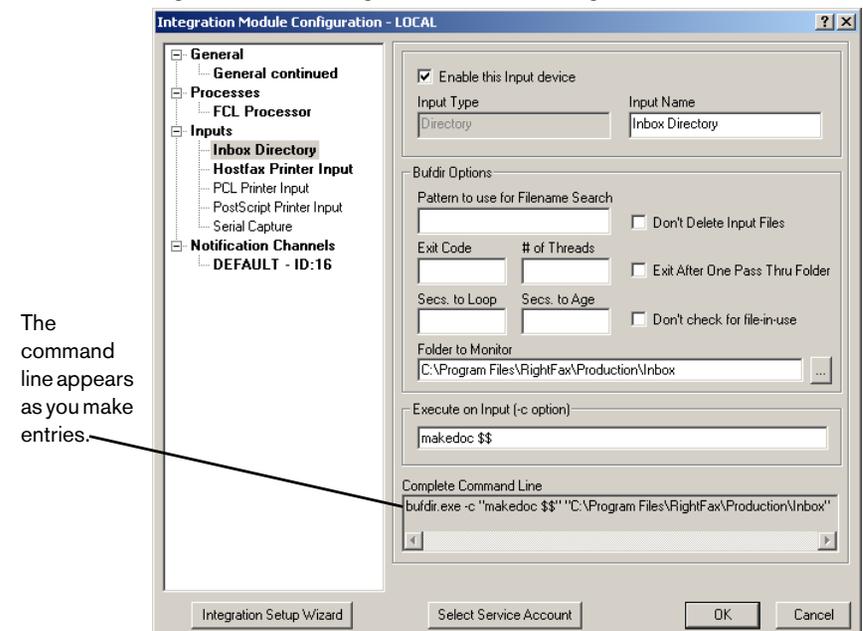
- In the **Input Type** list, select the type of data input needed for the communication method implemented for the host system. When you select an option, the available settings for the input appear in the **Integration Module Configuration** window.

- Enter the settings for the type of data input, as described in the following sections:

- [“Setting Up a Named Pipe Capture” page 27](#)
- [“Setting Up Directory Scanning” page 28](#)
- [“Setting Up a Serial Capture” page 29](#)
- [“Setting Up a TCP/IP Port Capture” page 31](#)
- [“Setting Up an IBM WebSphere MQ Connection” page 32](#)

To create the settings for an input, you will type a command line or select options in the dialog box. When you select an option in the dialog box, the option appears in the command line. For example, when you select the check box **Display Verbose Output**, **-v** appears in the **Complete Command Line** box.

Figure 6.3 The Integration Module Configuration Window



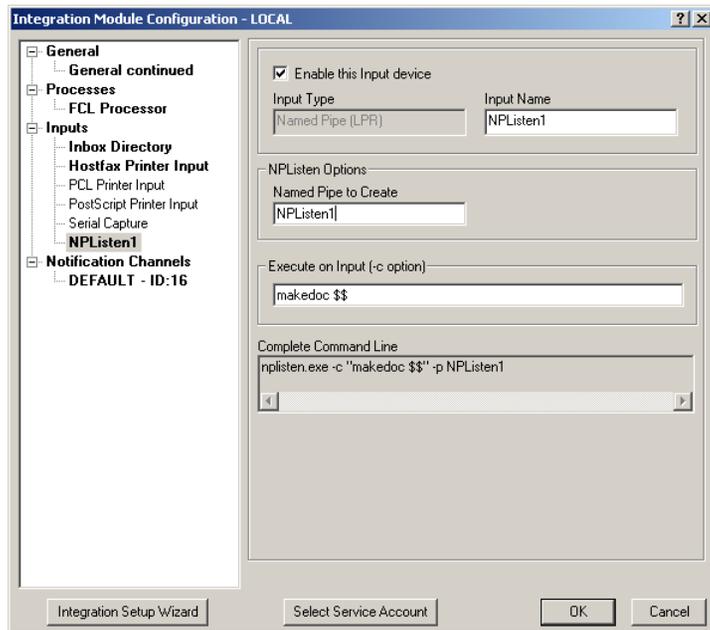
Setting Up a Named Pipe Capture

This procedure creates an input device that receives data from a host application via a named pipe. The program Nplisten.exe creates a named pipe, scans it, and then executes a command on data found in the pipe. Typically, this input type accommodates data that is formatted for a line printer (LPR).

To set up a named pipe capture

1. In the **Input Type** list, select **Named Pipe (LPR)**. The named pipe settings appear.

Figure 6.4 Settings for a Named Pipe Input Device



2. Select the check box **Enable this Input device**.
3. In the **Name** box, enter a descriptive name for this input device.

4. Complete the entries in the dialog box, as described in the following table. These settings correspond to Nplisten.exe command line options.

Table 6a Add Input Device Settings for a Named Pipe

Setting	Command line option	Description
Named Pipe to Create	-p <i>pipename</i>	The name of the named pipe that Nplisten.exe will monitor for files. You can enter any name for the pipe in one of the following formats: <ul style="list-style-type: none"> • pipename • \\.\pipe\pipename
Execute on Input (-c option)	-c "makedoc \$\$"	The command to execute on files received via the named pipe. Makedoc.exe begins the process of converting data from the host application into a fax. The variable \$\$ indicates that all files should be processed with Makedoc.exe.

Example nplisten.exe -c "makedoc \$\$" -p hplpr

In this example, Nplisten.exe will run the command "makedoc" on files received via the named pipe. The variable \$\$ indicates that all files will be processed with Makedoc.exe. "Hplpr" is a name for an LPR printer.

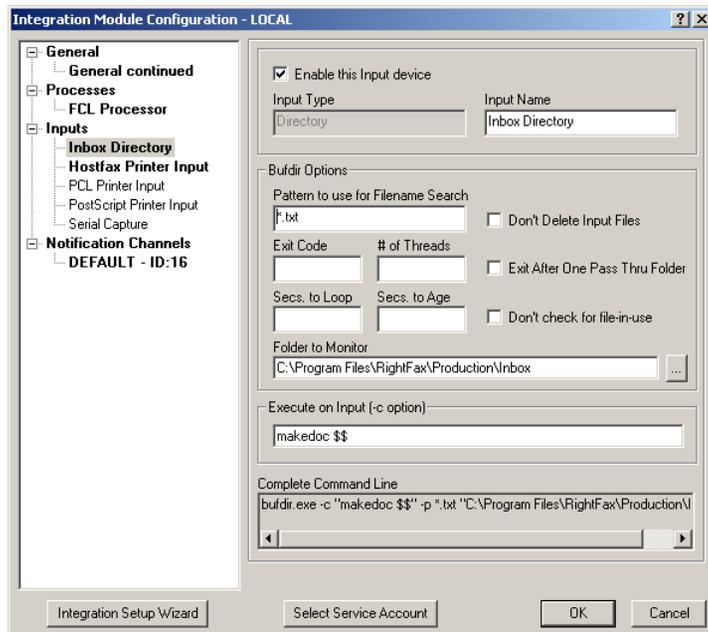
Setting Up Directory Scanning

This procedure creates an input device that retrieves data by scanning a folder for files. The program Bufdir.exe retrieves the files, creates a subdirectory for each calling fax server then copies the file into the subdirectory before they are processed. Thus, preventing duplication of faxes.

To set up directory scanning

1. In the **Input Type** list, select **Directory**. The directory settings appear.

Figure 6.5 Settings for a Directory Scanning Device



2. Select the check box **Enable this Input device**.
3. In the **Name** box, enter a descriptive name for this input device.

4. Complete the entries in the dialog box, as described in the following table. These settings correspond to Bufdir.exe command line options.

Table 6b Add Input Device Settings for Directory Scanning

Setting	Command line option	Description
Pattern to use for Filename Search	<code>-p pattern</code>	Enter the file types for which Bufdir.exe will scan, for example, *.txt. The default is *.*.
Don't delete Input Files	<code>-d</code>	This setting is useful when testing the connection to the host application. It determines whether or not the files in the folder are deleted after they are retrieved by Bufdir.exe. Files are deleted by default.
Exit Code	<code>-r code</code>	This setting determines that files will not be deleted until the program encounters a specified exit code.
# of Threads	<code>-t threads</code>	Specifies the maximum number of threads of Bufdir.exe to run simultaneously. The default is 1.
Exit after one pass through Folder	<code>-o</code>	Quit after scanning the folder once.
Secs. to Loop	<code>-l seconds</code>	This setting is commonly used with a shared folder on Windows. Enter the interval in seconds that Bufdir.exe will scan for files. The default is to scan the folder when notified by Windows that a file has been placed in the folder.

Table 6b Add Input Device Settings for Directory Scanning (Continued)

Setting	Command line option	Description
Secs. to Age	-w <i>seconds</i>	This setting assures that the file is up to date before it is processed. If your network is slow, this setting provides time for the completed document to be delivered to the folder. Enter the number of seconds to wait before retrieving the file.
Don't check for file in use	-u	This setting determines that Bufdir.exe will not check for files in use before retrieving them.
Folder to Monitor	Folder name and path	Enter the name of or the path to the folder to watch for documents.
Execute on Input (-c option)	-c "makedoc \$\$"	The command to execute on files received. Makedoc.exe begins the process of converting data from the host application into a fax. The variable \$\$ indicates that all files should be processed with Makedoc.exe.

Example bufdir -c "makedoc \$\$" c:\program files\rightfax\production\inbox

In this example, Bufdir.exe will run the command "makedoc" on files in the specified directory. The variable \$\$ indicates that all files will be processed with Makedoc.exe.

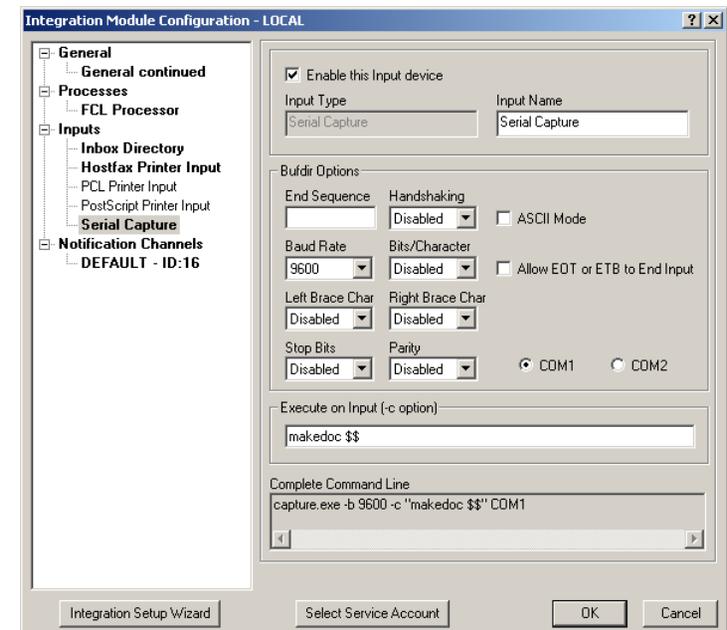
Setting Up a Serial Capture

This procedure creates an input device that retrieves data via a serial port. The program Capture.exe retrieves the data.

To set up a serial capture

1. In the **Input Type** list, select **Serial Capture**. The serial capture settings appear.

Figure 6.6 Settings for a Serial Capture Input Device



2. Select the check box **Enable this Input device**.
3. In the **Name** box, enter a descriptive name for this input device.

4. Complete the entries in the dialog box, as described in the following table. These settings correspond to Capture.exe command line options.

Table 6c Add Input Device Settings for a Serial Capture

Input setting	Command line option	Description
End Sequence	-S <i>sequence</i>	Specify the sequence of characters that will indicate the end of each document that is received via this serial port.
Handshaking	-H -x	Specify the handshaking method for the serial port. H = Hardware handshaking x = XON\XOFF (software) handshaking
Baud Rate	-b <i>baud rate</i>	The baud rate at which Capture.exe will scan the serial port.
Bits/Character	-C <i>size</i>	Specify the number of bits (7 or 8) per character.
ASCII Mode	-a	Select this check box to convert carriage returns in a document to carriage return-line feed pairs. In other words, <CR> (carriage return) will be converted to <CR><LF> (carriage return and line feed).
Left Brace Char	-L <i>character</i>	Specify one or more characters that form the left (opening) delimiter for FCL commands in documents that are received via this serial port. Usually this is a pair of left braces ({{}).

Table 6c Add Input Device Settings for a Serial Capture (Continued)

Input setting	Command line option	Description
Right Brace Char	-R <i>character</i>	Specify one or more characters that form the right (closing) delimiter for FCL commands in documents that are received via this serial port. Usually this is a pair of right braces (}}).
Allow EOT or ETB to end input	-E	Select this check box to specify that an end-of-text character (ASCII 4 or CTRL+D) or an embedded end-of-transmission-block (ETB) character (ASCII 23 or CTRL+W) will mark the end each document that is received via this serial port.
Stop Bits	-s <i>stopbits</i>	Specify the bit (0 or 1) that will indicate the end of each document that is received via this serial port.
Parity	-e -o	Specify the parity for this serial port. -e establishes even parity. -o establishes odd parity.

Table 6c Add Input Device Settings for a Serial Capture (Continued)

Input setting	Command line option	Description
COM1 or COM2	N/A	Specify the COM port. The default is COM1.
Execute on Input (-c option)	-c "makedoc \$\$"	The command to execute on files received. Makedoc.exe begins the process of converting data from the host application into a fax. The variable \$\$ indicates that all files should be processed with Makedoc.exe.

Example capture -b 9600 -c "makedoc \$\$" com1

In this example, Capture.exe will read input from the COM1 serial port and then execute the command "makedoc" on the data. The variable \$\$ indicates that all files will be processed with Makedoc.exe.

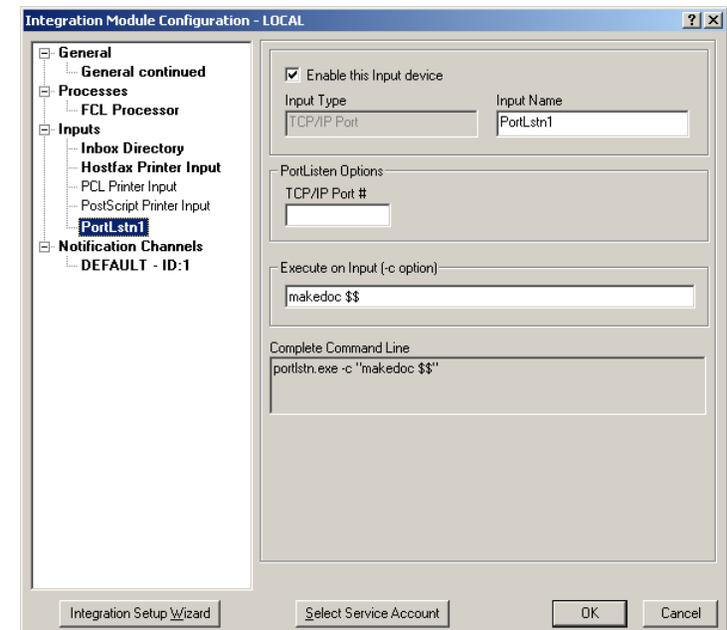
Setting Up a TCP/IP Port Capture

This procedure creates an input device that retrieves data via a TCP/IP port. The program PortLstn.exe retrieves the data.

To set up a TCP/IP port capture

1. In the **Input Type** list, select **TCP/IP Port**. The TCP/IP port settings appear.

Figure 6.7 Settings for a TCP/IP Port Input Device



2. Select the check box **Enable this Input device**.
3. In the **Name** box, enter a descriptive name for this input device.

4. Complete the entries in the dialog box, as described in the following table. These settings correspond to Portlstn.exe command line options.

Table 6d Add Input Device Settings for a TCP/IP Port Capture

Input setting	Command line option	Description
TCP/IP Port #	-p <i>number</i>	Enter the number of the port to monitor.
Execute on Input (-c option)	-c "makedoc \$\$"	The command to execute on files received. Makedoc.exe begins the process of converting data from the host application into a fax. The variable \$\$ indicates that all files should be processed with Makedoc.exe.

Example portlstn -c "makedoc \$\$" -p 6250

In this example, Portlstn.exe will read input from TCP/IP port 6250 and then execute the command "makedoc" on the data. The variable \$\$ indicates that all files will be processed with Makedoc.exe.

Setting Up an IBM WebSphere MQ Connection

This procedure describes the configuration needed to communicate with an IBM WebSphere MQ remote queue manager to retrieve messages (outgoing documents).

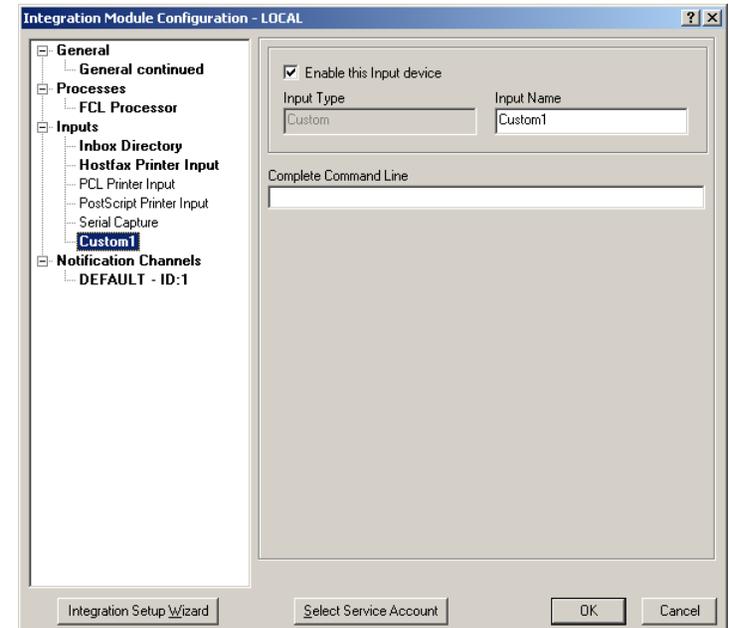
The program *mqget.exe* utilizes the IBM WebSphere MQ client to connect to the remote queue manager and retrieve messages from the specified remote queue. The communication input device is a

TCP/IP port. To configure the input, you specify the channel name, the host name, the queue manager, and the queue to retrieve the messages from.

To set up an IBM WebSphere MQ connection

1. In the **Input Type** list, select **Custom**. The custom settings appear.

Figure 6.8 Settings for a Custom Input Device



2. Select the check box **Enable this Input device**.
3. In the **Name** box, enter a descriptive name for this input device.

4. In the **Complete Command Line** box, enter a command. The Mqget.exe command line syntax and options are described in the following section.

Syntax **mqget -C channel -H hostname -M queuemanager -Q queue [options]**

Table 6e Mqget.exe Command Line Options

Option	Description
-C channel	Name to use for this connection channel.
-H hostname	Fully qualified domain name of the IBM WebSphere MQ queue manager.
-M queue manager	Queue manager for RightFax that is defined on the IBM WebSphere MQ server.
-Q queue	Queue to retrieve messages from. The Integration Module will monitor and retrieve messages from this queue.
-c "makedoc \$\$"	The command to execute on files received. Makedoc.exe begins the process of converting data from the host application into a fax. The variable \$\$ indicates that all files should be processed with Makedoc.exe.
-d	Display debugging output. This is helpful if you experience difficulty connecting to the server.
-p port	TCP/IP port number to use for remote connection. The default is 1414.
-1	Selects Version 1 of the WebSphere MQ Application Programming Reference. This option must be used because MQGet.exe is not designed to work with the WebSphere MQ API Version 2.
-tCCSID	Specifies the codeset name for a language. A list of the codeset IDs (CCSIDs) supported by WebSphere MQ is available from IBM.

Example `mqget -C RF_Chan -H qmmaster2 -M RightFax -Q RF_Queue -c "makedoc $$" -p 1414 -1`

In this example, Mqget.exe will monitor the IBM WebSphere MQ channel RF_Chan in the domain named Qmmaster2. It will connect to the queue manager RightFax and the queue named RF_Queue.

Data will be read via TCP/IP port 1414, and the command "makedoc" will be executed on the data. The variable \$\$ indicates that all files will be processed with Makedoc.exe.

Setting the MQGet polling interval

1. Log on to the RightFax server as an administrator.
2. Open the Windows registry editor and browse to HKLM\Software\RightFax\Production\MQSeries.
3. Create a new Dword entry called **MQ_Get_ConnectionTimeout**
4. Set the data value to the interval for scanning the queue, in seconds.
5. Close Windows registry editor

■ ■ ■

Chapter 7

Testing a Connection and an Input Device

This chapter describes methods for testing that the RightFax server, Integration Module, and host application are connected and communicating.

Testing a Named Pipe Connection

This procedure tests the connection from the host application to a print input device on the RightFax server, as described in [“Setting Up a Named Pipe Capture”](#) on [page 27](#).

To test the connection

1. Pause printing to the printer that was created for the named pipe input.
2. Temporarily change the printer port to File.
3. Send a test document from the host application to the printer.
4. Look in the print queue to verify that the test document has been sent to print.
5. Activate printing to the Integration Module printer, specifying the file name for the document.

6. Open and examine the test document.
 - Documents that are created with filter templates (filter mode) will be plain ASCII text.
 - Documents that are created in native mode will be FCL documents—a combination of FCL commands and document data from the host application.
 - PCL or PostScript documents will be in PCL or PostScript format with FCL commands on the first “false” page.
7. Change the printer port back to the named pipe.

Testing Directory Scanning

This procedure tests the connection from a host system to a folder that is scanned for files by the Bufdir.exe program, as described in [“Setting Up Directory Scanning”](#) on [page 28](#).

To test the connection

1. Stop Bufdir.exe by stopping the Integration Module service in Enterprise Fax Manager.
2. Send a test document from the host application to the folder specified in the Bufdir.exe command line.
3. Look in the target folder on the Integration Module to verify that the test document was received.

4. Start the Integration Module service in Enterprise Fax Manager.

Testing a Serial Connection

This procedure tests the connection from the host system to a serial port on the RightFax server, as described in [“Setting Up a Serial Capture”](#) on page 29.

To test the connection

1. Stop Buffer.exe using one of the following methods:
 - Stop it from the **Process** tab in Windows Task Manager.
 - Enter **signal HFBufferStop** at the command prompt.
 - Enter **kill /f buffer.exe** at the command prompt (not recommended).
2. Verify that Buffer.exe is stopped by looking in Windows Task Manager.
3. Send a test document from the host application to the serial port specified in the Capture.exe command line.
4. In Windows Explorer, navigate to RightFax\Production\Makedoc and verify that the test document was received.
5. In Enterprise Fax Manager, start Buffer.exe by stopping and starting the Integration Module service.

Testing a TCP/IP Port Connection

This procedure tests the connection from the host application to a TCP/IP port on the RightFax server, as described in [“Setting Up a TCP/IP Port Capture”](#) on page 31

To test the connection

1. Stop Buffer.exe using one of the following methods:
 - Stop it from the **Process** tab in Windows Task Manager.
 - Enter **signal HFBufferStop** at the command prompt.
 - Enter **kill /f buffer.exe** at the command prompt (not recommended).
2. Verify that Buffer.exe is stopped by looking in Windows Task Manager.
3. Send a test document from the host application to the TCP/IP port specified in the Portltn.exe command line.
4. In Windows Explorer, navigate to RightFax\Production\Makedoc and verify that the test document was received.
5. In Enterprise Fax Manager, start Buffer.exe by stopping and starting the Integration Module service.

Testing an IBM WebSphere MQ Connection

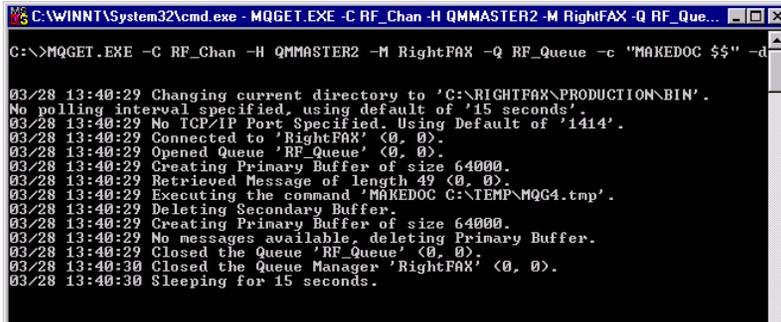
This procedure tests the connection from the host system to an IBM WebSphere MQ queue on the RightFax server, as described in [“Setting Up an IBM WebSphere MQ Connection”](#) on page 32.

To test the connection

1. Open a command prompt window.
2. At the command prompt, enter the command line that was written to create the IBM WebSphere MQ input, and then press ENTER.

If the connection is successful, then you should see text similar to that shown in the following example. If the input connection is not successful, then you will see error messages.

Figure 7.1 Successful Test of IBM WebSphere MQ Connection



```
C:\WINNT\System32\cmd.exe - MQGET.EXE -C RF_Chan -H QMMASTER2 -M RightFAX -Q RF_Queue -c "MAKEDOC $$$" -d
C:\>MQGET.EXE -C RF_Chan -H QMMASTER2 -M RightFAX -Q RF_Queue -c "MAKEDOC $$$" -d
03/28 13:40:29 Changing current directory to 'C:\RIGHTFAX\PRODUCTION\BIN'.
03/28 13:40:29 No polling interval specified, using default of '15 seconds'.
03/28 13:40:29 No TCP/IP Port Specified, Using Default of '1414'.
03/28 13:40:29 Connected to 'RightFAX' (0, 0).
03/28 13:40:29 Opened Queue 'RF_Queue' (0, 0).
03/28 13:40:29 Creating Primary Buffer of size 64000.
03/28 13:40:29 Retrieved Message of length 49 (0, 0).
03/28 13:40:29 Executing the command 'MAKEDOC C:\TEMP\MQG4.tmp'.
03/28 13:40:29 Deleting Secondary Buffer.
03/28 13:40:29 Creating Primary Buffer of size 64000.
03/28 13:40:29 No messages available, deleting Primary Buffer.
03/28 13:40:29 Closed the Queue 'RF_Queue' (0, 0).
03/28 13:40:30 Closed the Queue Manager 'RightFAX' (0, 0).
03/28 13:40:30 Sleeping for 15 seconds.
```

■ ■ ■

Chapter 8

Creating FCL Documents

To create documents with FCL, you must insert FCL commands into the data stream that comes from the host application. When the Integration Module receives this data stream, it uses the FCL to format and send the document.

Because of the range of host-based applications, system environments, and document formats, specific instructions for accessing and manipulating document data is beyond the scope of this guide.

About 100 FCL commands are available. For a detailed list see [Appendix B, "FCL Commands"](#).

Each FCL document must have at least three commands: `{{begin}}`, `{{end}}`, and a command that specifies the type of transmission. An overview of these commands is provided in the following table.

Table 8a Required FCL Commands

Command	Description
<code>{{begin}}</code>	<p>Indicates the beginning of a document. The Integration Module will process all the data that appears between a <code>{{begin}}</code> and an <code>{{end}}</code> command as a discreet document. Data that does not appear between the <code>{{begin}}</code> and <code>{{end}}</code> commands is ignored.</p> <p>This command must appear as the first command in each FCL document. This command is required in every FCL document. FCL documents may contain multiple <code>{{begin}}</code> and <code>{{end}}</code> commands, but the FCL between each set of commands will be rendered as a separate page.</p>
<code>{{end}}</code>	<p>Indicates the end of a document. The Integration Module will process all the data that appears between a <code>{{begin}}</code> and an <code>{{end}}</code> command as a discreet document. Data that does not appear between the <code>{{begin}}</code> and <code>{{end}}</code> commands is ignored.</p> <p>This command must appear as the last command in each FCL document. FCL documents may contain multiple <code>{{begin}}</code> and <code>{{end}}</code> commands, but the FCL between each set of commands will be rendered as a separate page.</p>

Table 8a Required FCL Commands (Continued)

Command	Description
<code>{{fax}}</code>	<p>Contains the fax number for the recipient and specifies that the document will be sent via fax.</p> <p>If you have licensed the InternetLink Module, then you can specify an e-mail address rather than a fax number in order to e-mail documents.</p>
<code>{{type certified}}</code>	Specifies that the document will be sent as a RightFax SecureDocs certified delivery e-mail message. To use this option, you must license and install the RightFax SecureDocs Module.
<code>{{type e-mail}}</code>	Specifies that the document will be sent as the body of an e-mail message. To use this option, you must license the RightFax InternetLink Module.
<code>{{type file}}</code> <code>{{file}}</code> <code>{{list}}</code>	Specifies that the document will be transferred to a folder.
<code>{{type mime}}</code>	Specifies that the document will be sent as an attachment to an e-mail message. This option requires the RightFax InternetLink Module.
<code>{{type print}}</code>	Specifies that the document will be sent to a printer.
<code>{{type SMS}}</code> <code>{{smsmsg MessageText}}</code> <code>{{sms PhoneNum}}</code>	Specifies that a message will be sent to an SMS-compatible device.

Understanding the Format of FCL Commands

The most important requirement for forming FCL commands is that they begin with two left braces ({{) and end with two right braces (}}). Unique delimiters can be defined for serial capture input. For instructions, see [“Setting Up a Serial Capture”](#) on page 29.

All FCL commands have the same format, and most of them can include arguments and variables:

```
{{command argument variable}}
```

Use one or more spaces to separate commands from arguments and variables.

You can list several FCL commands in a sequence. Do not type spaces between the commands in a line:

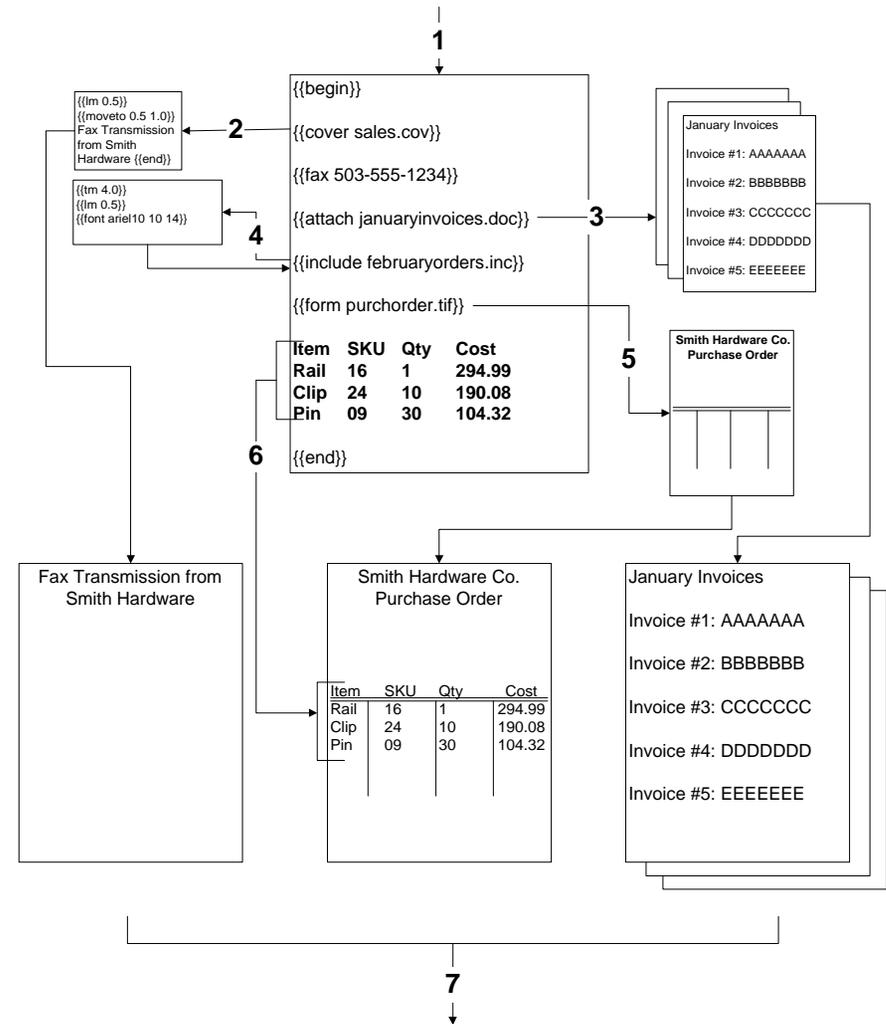
```
{{begin}}{{fax 503-555-1234}}{{onsuccess fax 555-6892}}
```

You can place FCL commands anywhere in a document, but Captaris recommends that you place them at the end unless otherwise specified for the FCL command (see [Appendix B, “FCL Commands”](#)).

Example FCL Documents

The following figure ([“Figure 8.1: Other Files Can Be Added to the Document”](#)) and the numbered paragraphs following it illustrate how other documents—such as cover sheets, background forms, and attachments—can be included in a document with the use of FCL commands.

Figure 8.1 Other Files Can Be Added to the Document



1. The Integration Module receives the document data stream from the host. It contains the source document data and FCL commands. In the example here, the original host data is shown in bold text. It will become part of a purchase order. The FCL commands precede and follow the host data.
2. The `{{cover}}` command specifies that the file Sales.cov will be used as the fax cover sheet. The example cover sheet creates a 0.5-inch left margin (`{{lm}}`) and starts the text “Fax Transmission from Smith Hardware” at the x- and y-coordinates 0.5 and 1.0 inches from the top-left corner. The cover sheet is attached as the first page of the document.
3. The `{{attach}}` command specifies that the file Januaryinvoices.doc will be attached. Januaryinvoices.doc will be attached following the body of the fax.
4. The `{{include}}` command specifies that the file Februaryorders.inc will be referenced. The include file specifies the measurements for top and left margins (`{{tm}}` and `{{lm}}`) and the font.
5. The `{{form}}` command specifies that the file Purchorder.tif will be added as the background for the fax.
6. The finished fax includes the background form and the document data, as if the document has been printed on a pre-printed form.
7. The cover sheet, document, and attachment are faxed to 503-555-1234 in one transmission.

The following figure (“[Figure 8.2: FCL Document for an Invoice from NW Distributors to Portland Trading Co.](#)”) illustrates the FCL-encoded document for an invoice. The finished document is illustrated in “[Figure 8.3: Finished Invoice from NW Distributors to Portland Trading Co.](#)”.

Figure 8.2 FCL Document for an Invoice from NW Distributors to Portland Trading Co.

```

1 →  {{begin}}
2 →  {{fax 503 555 9182}}
3 →  {{form invoice.tif}}
4 →  {{attach ptc6mos.xls}}
5 →  {{tm 1.25}}
6 →  {{lm 0.25}}
   {{font "times new roman" 12}}
   {{orient landscape}}
   {{company Portland Trading Co.}}
   {{contact Anne Paige}}
   {{owner John Bardolph}}
   {{onerror fax 503-555-4592}}
   {{notifyhost notifyinv notifyinv exchange}}
   18277                               8/8/01

Anne Paige                                Portland Trading Co.
Portland Trading Co.                      11990 S.W. Capitol Hill Road
11990 S.W. Capitol Hill Road              Portland, OR 97034
Portland, OR 97034                        (503) 555-9182

55222      13579                          John Bardolph
8/4/01      8/7/01                          2% 10 days

50 lb  13654  Beef Jerky                    $ 8.50/lb    $ 425.00
50 lb  13655  Smoked Salmon                 $ 12.00/lb   $ 600.00

                                           $1,025.00   $75.00    $75.00    $1,175.00

{{end}}

```

1. The `{{form}}` command specifies that the file Invoice.tif will be added as the background for the fax. “[Figure 8.3: Finished Invoice from NW Distributors to Portland Trading Co.](#)” shows the finished document merged with the Invoice.tif background form.
2. The `{{attach}}` command retrieves a file named Ptc6mos.xls. This file is a six-month history of items purchased by Portland Trading Company. It is a Microsoft® Excel spreadsheet that is populated with data from a database. The spreadsheet will be sent as an attachment.

3. These four commands format the document. `{{Tm}}` specifies a top margin of 1.25 inches. `{{Lm}}` specifies a left margin of 0.25 inches. `{{Font}}` specifies a font of Times New Roman, 12-point. `{{Orient}}` specifies that the document will be in landscape orientation.
4. These three commands will populate the variables in a notification message that will notify the sender when the document is sent.
5. The `{{onerror}}` command specifies that the Integration Module take a particular action if a document fails to send properly. If an error occurs in transmission, the document will be faxed to 503-555-4592, which is a fax machine in the sender's office.

6. The `{{notifyhost}}` command specifies that the sender will receive a notification message when the document is sent.

Figure 8.3 Finished Invoice from NW Distributors to Portland Trading Co.

Page 1 of 4

ptc6mos.xls

Data Data Data Data Data Data Data Data

NW Distributors		Invoice # 18277	Date 8/8/01	
Sold to Anne Paige Portland Trading Co. 11990 S.W. Capitol Hill Road Portland, OR 97034		Ship to Portland Trading Co. 11990 S.W. Capitol Hill Road Portland, OR 97034 (503) 555-9182		
55222 Your order #	13579 Our order #	John Bardolph Sales person		
8/4/01 Date ordered	8/7/01 Date shipped	2% 10 days Terms		
Qty	Item	Description	Price	Amount
50 lb	13654	Beef Jerky	\$ 8.50/lb	\$ 425.00
50 lb	13655	Smoked Salmon	\$ 12.00/lb	\$ 600.00
Subtotal		Sales tax	Shipping	Total
\$1,025.00		\$75.00	\$75.00	\$1,175.00

■ ■ ■

Chapter 9

Setting Defaults for FCL Documents

To set most default values, you must make configurations to the Integration Module Configuration program. In many cases, you can override the default in specific documents by including FCL commands in the document data from the host application.

You can also set defaults for things like margins and fonts by inserting relevant FCL commands in a global include file. For information on include files, see [Chapter 13, “Including the Same Files and Commands in Many Documents”](#).

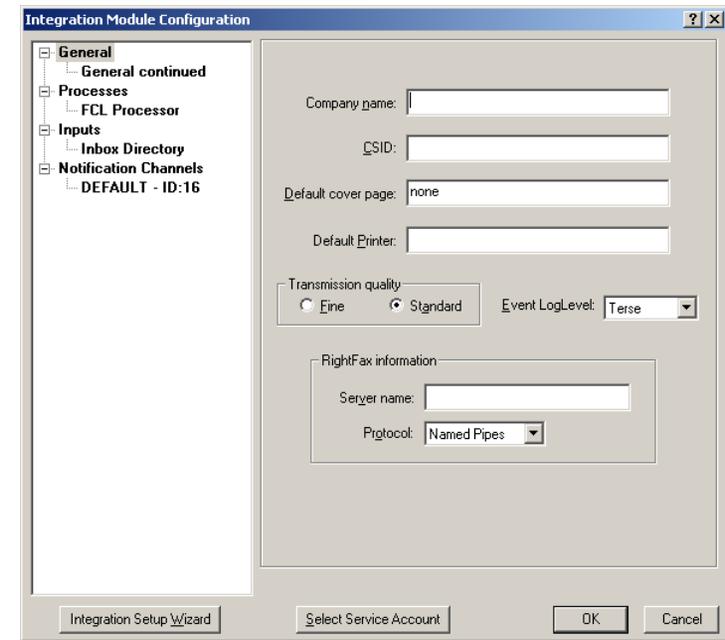
Opening the Integration Module Configuration Program

1. On the **Start** menu, select **Programs > RightFax > Enterprise Fax Manager**. The **Enterprise Fax Manager** window appears.
2. In the **Fax Servers** list, click the name of the server on which the Integration Module is running.
3. In the **Service Name** list, double-click **RightFax Integration Module**. The **Integration Module Configuration** window appears.

Setting General Defaults

1. In the left pane, click **General**. The **General** settings appear in the right pane.

Figure 9.1 General settings



2. In the **Company name** box, enter a name to appear on the cover sheet of each sent fax. Usually this is the company name.

This setting can be overridden for a document with the `{{rti}}` command. For more information on this command, see [Appendix B, "FCL Commands"](#).

3. In the **CSID** box, enter a voice telephone number to appear on the cover sheet of each sent fax. This is usually the telephone number for the company.

This setting can be overridden for a document with the `{{csi}}` command. For more information on this command, see [Appendix B, "FCL Commands"](#).

4. In the **Default cover page** box, enter the file name for a cover sheet. Enter **none** for no cover sheet. For more information on cover sheets, see [Chapter 14, "Creating and Attaching Cover Sheets"](#).

5. In the **Default Printer** box, specify the printer to use for printing documents when an error is encountered or when printing a document with the `{{type print}}` FCL command. Enter a printer that has been defined in Enterprise Fax Manager.

This setting can be overridden for a document with the `{{printer}}` command. For more information on this command, see [Appendix B, "FCL Commands"](#).

6. Under **Transmission quality**, specify the image quality for faxes.
 - Fine is 200 x 200 dots-per-inch.
 - Standard is 200 x 100 dots-per-inch.

This setting can be overridden for a document with the `{{quality}}` command. For more information on this command, see [Chapter 18, "Setting Page Orientation and Image Quality"](#).

7. In the **Event Log Level** box, select the level of reporting to the Windows Event Log.

- None records no errors
- Terse records critical errors only
- Normal records the most significant events only and is the correct setting for normal use
- Verbose records all significant events and is the most useful for tracking and resolving problems



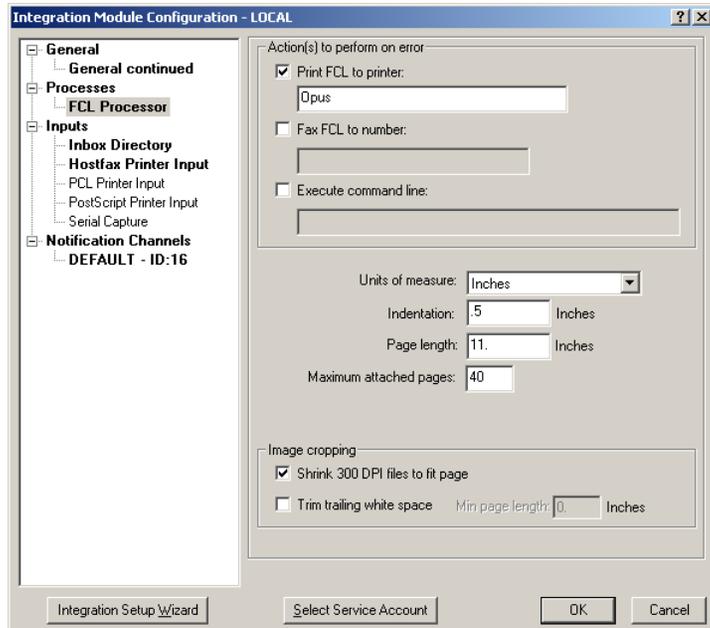
Caution *If this value is set to verbose indefinitely, the event log can become full. This can prevent new events from being logged.*

8. In the **Server name** box, enter the name of the RightFax server.
9. In the **Protocol** list, select the communication protocol for the RightFax server.

Setting Defaults for FCL Processing

1. In the left pane, click **FCL Processes**. The **FCL Processor** settings appear in the right pane.

Figure 9.2 The FCL Processor Settings



2. In the **Units of measure** box, specify the unit of measurement that will be used for indentation, page length, and for FCL commands. With FCL commands, this setting is used to specify the x- and y-coordinates for placing data on a page. This setting can be overridden for a document with the `{{units}}` command. For more information on this command, see [Appendix B, "FCL Commands"](#).
3. In the **Indentation** box, specify the indentation of the fax image on the left side of the page. The unit of measurement is determined by the **Units of measure** setting.

4. In the **Page length** box, enter the maximum page length for a fax. For example, this setting prevents a legal-size page (11 inches wide x 14 inches long) from being cut off at 11 inches. The unit of measurement is determined by the **Units of measure** setting.
5. In the **Maximum attached pages** box, specify the maximum number of attached pages for each fax. The Integration Module can attach up to 1024 pages to a fax.
6. A document attached to a fax that is 300 dots-per-inch will be faxed as two pages. To reduce the image size and send the image as one page, select the check box **Shrink 300 DPI files to fit page**.
7. In the **Minimum page length** box, specify the minimum page length for faxes. This setting ensures that a fax with very little data is printed on a page that is long enough, so that it does not get lost or discarded by the recipient. The unit of measurement is determined by the **Units of measure** setting.
8. White space at the end of a fax is sent as data to the receiving fax machine. You can speed up the transmission of the fax if this data is not sent. To remove this data from the fax, select the **Trim trailing white space** check box.

■ ■ ■

Chapter 10

Creating Filter Templates

Understanding Filter Templates

The RightFax Integration Module processes documents from the host application by interpreting facsimile command language (FCL) and performing functions based on the commands. The Integration Module can do this in one of two ways: native mode or filter mode.

- With native mode, you include FCL commands in the document data that is sent from the host application. This may require custom programming to add FCL to documents or to templates in the host application or to insert FCL in the data stream.
- With filter mode, you create “filter templates” that add FCL to the document data after it is sent to the Integration Module for processing. A filter template is a map of the document data that contains the FCL that is required to create and send the document.

This chapter discusses the creation of filter templates that support filter mode.

Filter templates provide the following features:

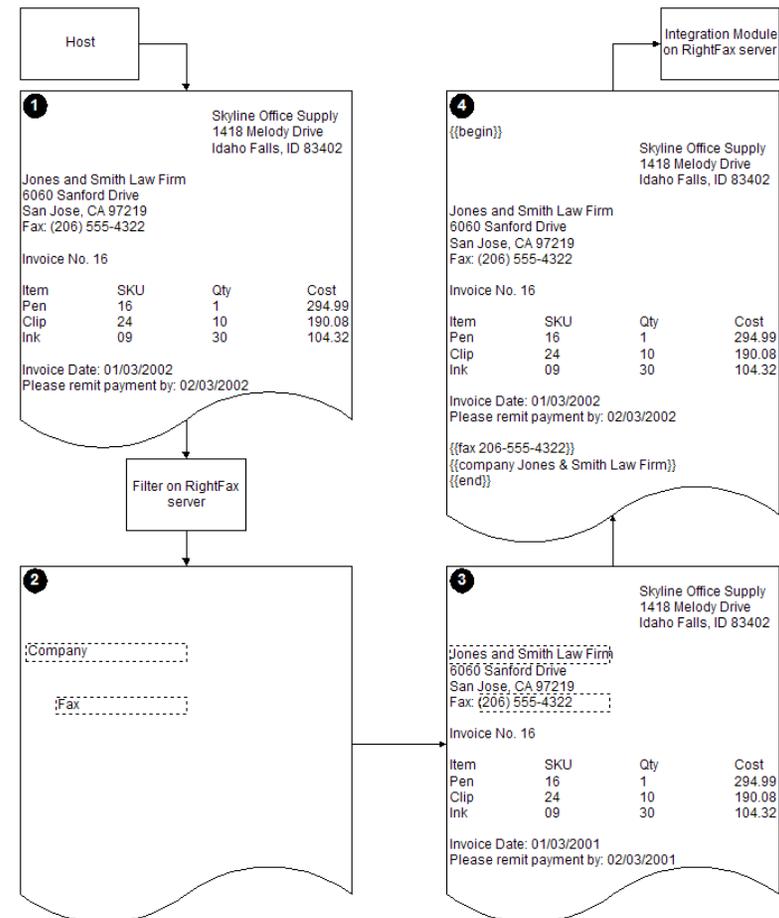
- Data mapping provides sending information to the RightFax Integration Module, such as the recipient name, fax number, and e-mail address. It also provides information from the source document for the fax cover sheet.
- Data mapping provides information from the source document for notification messages, such as the sender's name and e-mail address. Combined with the RightFax Integration Module, notifications provide the status of the sent document to the sender, to a system administrator, or another recipient.
- Background forms can provide visual interest to the filter template, with the features of a pre-printed form such as a purchase order or stationery. Background forms are included by linking an image file to a filter template.
- Graphics can be added. Graphics are included by linking an image file to a filter template.
- Data tables can be linked to the form so that information can be added to a document before it is sent. This is useful for adding data that is not sent from the host application with the document.
- User-defined data fields and facsimile command language (FCL) can further extend the filter template.

In the following example, the mapped data elements are the company name and fax number.

1. An invoice document is sent from host application that does not include FCL.
2. On the RightFax server, Filter.exe receives the document data and it retrieves a filter template. The filter template identifies data in the document that is needed for addressing and sending the document. In this filter template, the company name and fax number are mapped.
3. The filter software extracts the content of the company name and fax number fields and creates the `{{company}}` and `{{fax}}` FCL commands.

4. The document data now contains FCL. The `{{company}}` and `{{fax}}` FCL commands are created by the mapped fields in the filter template. The document can now be processed and sent by the Integration Module.

Figure 10.1 Filter Mode Document Flow



Preparing to Create Filter Templates

Before you begin creating filter templates, prepare the following information:

- For each document that will be sent, obtain an ASCII text file that includes the data that will be used to send the document, to send a notification message, or to be placed on cover sheets. This data might include the fax number, printer, voice telephone number, and the sender's name and address. Save the text files in the folder RightFax\Production\Include.

To capture a document from the host application data stream, send the document via an input channel on the Integration Module. The host data must be ASCII text with no extra characters or encoding. This may mean that a customized script must be written to prepare the data for the filter template. The file format should not be PCL, PostScript, or Portable Document Format (PDF).

- Identify each background form that should be included with each document. Create the forms and save them in the appropriate folder on the RightFax server, as described in [Chapter 12, "Creating and Linking Background Forms"](#).
- Identify each cover sheet that should be included with each document. Create the cover sheets and save them in the appropriate folder on the RightFax server, as described in [Chapter 14, "Creating and Attaching Cover Sheets"](#).
- Identify each graphic element, such as logos or signatures, to include in the filter template. Create the graphics and save them in the appropriate folder on the RightFax server, as described in [Chapter 19, "Attaching and Embedding Files, Signatures, and Graphics"](#).
- Create include files, if needed. For more information, see [Chapter 13, "Including the Same Files and Commands in Many Documents"](#).
- Create lookup tables, if needed. For more information, see the {{lookup}} FCL command in [Appendix B, "FCL Commands"](#).

- Create TCP/IP print queues in the Integration Module. This is typically one print queue per filter.

Assigning filter templates to documents

You have two options for assigning a filter template to a document:

- Assign a template ID to each filter template. Data in the document from the host application is mapped to a field in the filter template that identifies the template (see ["Understanding the Field Types"](#) on [page 59](#)).
- Create up to 70 unique input channels for the documents to process with filter templates. One input channel can process documents for one filter template (see ["Creating an Input Channel for Filter Documents"](#) on [page 64](#)).

Maintaining the filter template

Filter templates are designed to work with data that occurs in static locations in the document data. If the document changes and a data element is moved to a new location, then the filter template must be revised to fit the change.

Creating a New Filter Template

You use the MapText program to create filter templates.

MapText includes a preview function, so that you can preview and adjust each template as you create it.

To create a filter template, complete the following steps:

["Step 1: Overlay the sample data on a MapText document"](#) ([page 52](#))

["Step 2: Map the document data to MapText fields"](#) ([page 54](#))

["Step 3: Establish page length, orientation, and background forms"](#) ([page 57](#))

["Step 4: Insert FCL"](#) ([page 57](#))

[“Step 5: Preview the filter” \(page 58\)](#)

The instructions in this chapter refer to sample files that are installed with the Integration Module. The files are described in the following table.

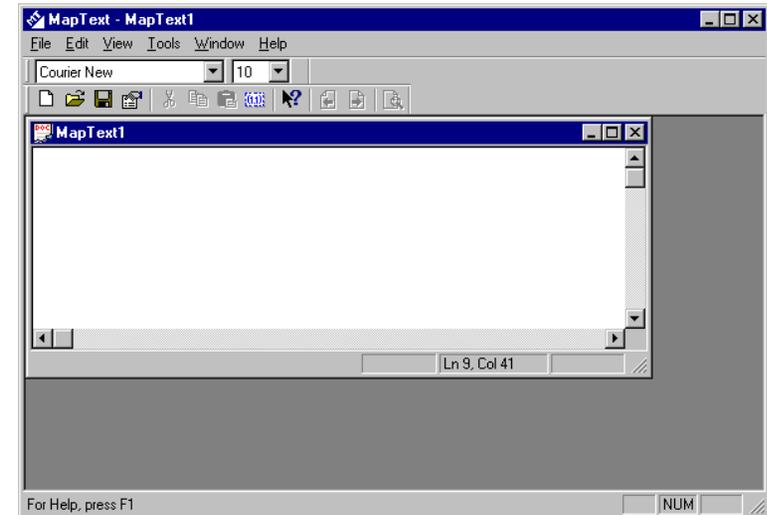
Table 10a Sample Files Installed with Filter for Production

Sample file	Location	Description
SampleData.txt	RightFax\Production\Include	An example of a document that a host application might create. It is formatted as a single print stream capture that contains two documents: a three-page purchase order and a one-page purchase order.
SampleForm.tif	RightFax\Production\Forms	An example of a form onto which you might print a purchase order. MapText will merge SampleData.txt with SampleForm.tif to create an image of the document printed on a pre-printed form.

Step 1: Overlay the sample data on a MapText document

1. On the RightFax server in Windows Explorer, navigate to RightFax\Production\Bin and run MapText.exe. The MapText window opens with a blank document called MapText1.

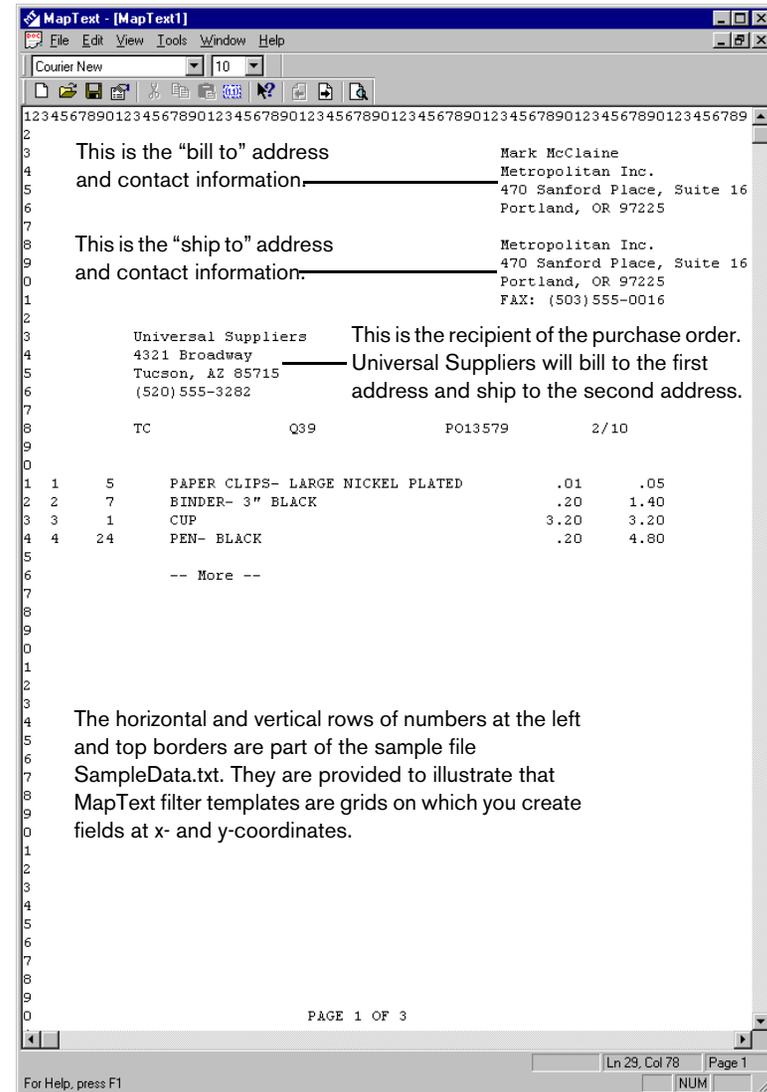
Figure 10.2 The MapText Window



2. On the **Tools** menu, click **Overlay Data File**.

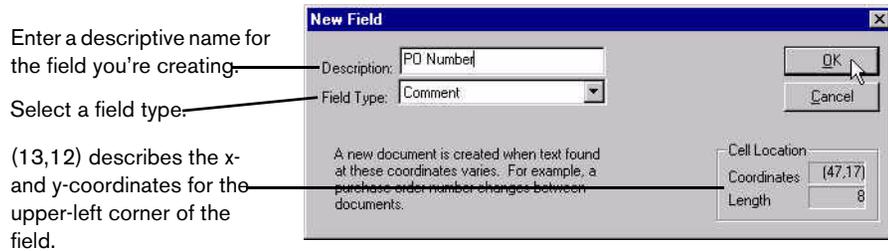
3. Select your sample document or SampleData.txt, and then click **Open**. The following example illustrates that SampleData.txt is placed in the blank MapText document. This image shows the data that forms Metropolitan Inc. purchase orders. In this case, the purchase order will go to Universal Suppliers.

Figure 10.3 Host Data Merged with a Blank MapText Document



- In the **New Field** dialog box, in the **Description** box, enter a description for the field, such as “Company Name.”

Figure 10.5 Assigning Attributes to the Field

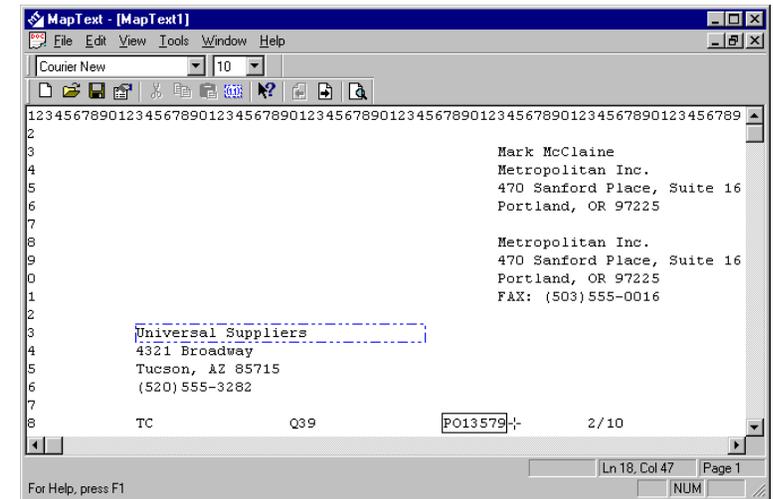


- In the **Field Type** list, click **Company**.
- Click **OK**.

- Define the field in the document template that will contain the purchase order number. Drag the mouse to select the purchase order number. The **New Field** dialog box appears.

In the following figure, the purchase order number (PO 13579) is selected.

Figure 10.6 Defining the Purchase Order Number as a Field

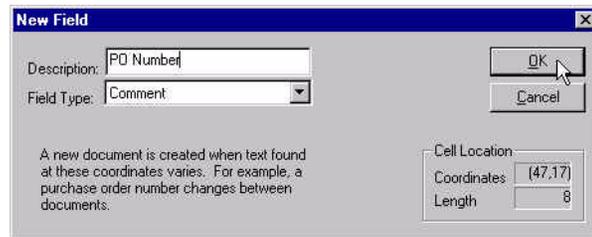


- In the **Description** box, enter a description for the field, such as “PO Number.”

7. In the **Field Type** list, click **Comment**.

One attribute of the Comment field type is that it starts a new document when the data in the field changes. In this case, a new purchase order document will be created with each new purchase order number that is received. For detailed information on the Comment field type, see [“Understanding the Field Types”](#) on page 59.

Figure 10.7 Mapping the Purchase Order Field

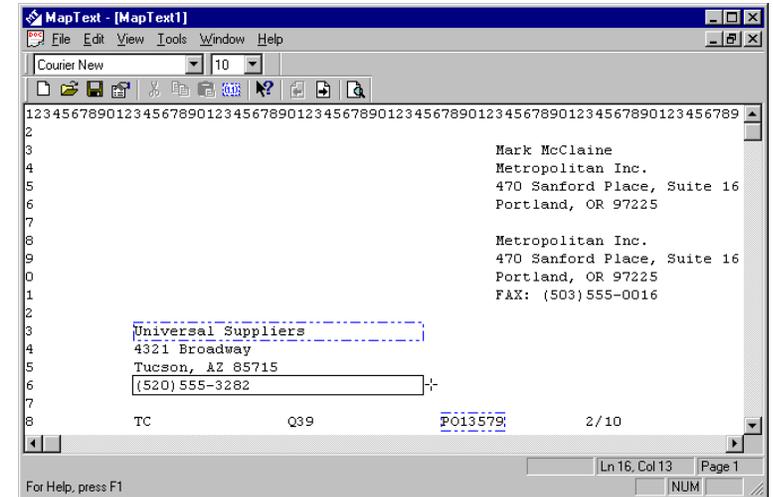


8. Click **OK**.

9. Define the field in the document template that will contain the fax number of the company that will receive this document. Drag the mouse to select the fax number. The **New Field** dialog box appears.

In the following figure, the fax number (520-555-3282) is selected.

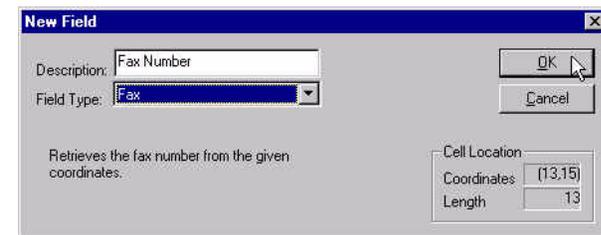
Figure 10.8 Defining the Recipient Fax Number as a Field



10. In the **New Field** dialog box, in the **Description** box, enter a description for the field, such as “Fax Number.”

11. In the **Field Type** list, click **Fax**.

Figure 10.9 Mapping the Fax Field Type



12. Click **OK**.

13. On the **File** menu, click **Save**.

14. Give the filter template a descriptive name. It will be saved as a MapText document (.mtd) in the RightFax\Production\Include folder.

In this example, you have:

- Mapped the company name in purchase order documents to the “Company Name” field of the template. This created the FCL command `{{company Universal Suppliers}}` that RightFax will use to address the fax.
- Mapped the purchase order number in purchase order documents to the “Purchase Order” field of the template. This created the FCL command `{{comment PO13579}}`.
- Mapped the fax number in purchase order documents to the “Fax Number” field of the template. This created the FCL command `{{fax 520 555 3282}}` that RightFax will use to send the fax.
- Saved the purchase order template.

When RightFax receives a document using this filter template, it will extract data from the defined fields and use it to address and send the fax. It will also begin a new document each time a new purchase order number is encountered in the purchase order field.

Step 3: Establish page length, orientation, and background forms

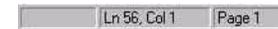
1. On the **File** menu, click **Properties**, or click the **Properties** button  on the toolbar. The **Properties** dialog box appears.
2. In the **Form** list, select `SampleForm.tif`, or select a background form that you have created.

3. In the **Page Length (lines)** box, enter **56**.

`SampleData.txt` includes an end-of-page symbol that indicates the bottom of the second page. The page breaks at whichever comes first: the number you enter in the **Page Length (lines)** box or an end-of-page symbol. In `SampleData.txt`, the end-of-page symbol is a form feed symbol (^L, or ASCII-12).

If the data from the host application does not include an end-of-page symbol, you must specify the page length. To do this in MapText, point the mouse at the last line on a page, and then look at the status area in the lower-right corner of the MapText window. The following figure shows that `SampleData.txt` is 56 lines long.

Figure 10.10 The MapText Tray



The default page length for documents is 66 lines for portrait orientation and 33 lines for landscape orientation. You can test the validity of the count in “[Step 5: Preview the filter](#)” on [page 58](#).

4. Select **Portrait**.

5. In the **Print copies in absence of fax number** box, click **1**. When no fax number appears in the document from the host application, this function sends the specified number of copies to the default printer specified for the Integration Module.

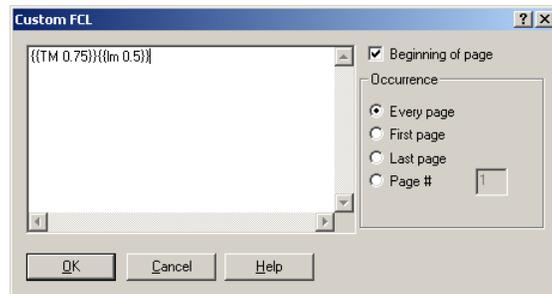
Step 4: Insert FCL

You can extend the filter template by adding FCL commands. For a description of the FCL commands, the proper syntax, and the placement of the commands within documents, see [Appendix B](#), “[FCL Commands](#)”.

To insert FCL

1. On the **File** menu, click **Properties**, or click the **Properties** button  on the toolbar. The **Properties** dialog box appears.
2. Click **New**. The **Custom FCL** dialog box appears.

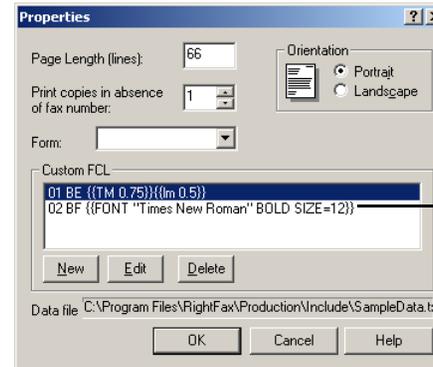
Figure 10.11 The Custom FCL Dialog Box



3. Enter the FCL using the proper syntax and delimiters.
4. To specify that the commands will be inserted at the beginning of each document, select the **Beginning of page** check box.
5. Select one of the following options:
 - To specify that the FCL should apply to every page of the document, click **Every page**.
 - To specify that the FCL should apply to only the first page of the document, click **First page**.
 - To specify that the FCL should apply to only the first page of the document, click **Last page**.
 - To specify the page number for the FCL, click **Page #**, and enter the page number.
6. Click **OK**. The MapText window appears.

7. On the **File** menu, click **Save**, or click the **Save** icon on the toolbar. The following example illustrates how FCL appears in the **Properties** dialog box after it is defined.

Figure 10.12 Custom FCL Added to the Filter



“01” indicates that this is the first line of FCL code in this filter template, and so on.

“BE” indicates that this line of FCL will apply to the beginning of every page of the document that uses this filter template.

“BF” indicates that this line of FCL will apply to the beginning of the first page of the document that uses this filter template.

Step 5: Preview the filter

To preview the filter, use the Generate Image function. Generate Image displays the background form merged with the sample document data and therefore helps you fine-tune the alignment. It does not test that the RightFax server can receive data from the host application and process the data correctly with the filter.

To preview the filter

On the **Tools** menu, click **Generate Image**, or click the **Generate Image** button  on the toolbar. MapText generates the image, and the default TIFF image viewer opens with the finished document.

Examine all the pages of the finished document. If you are using a background form, verify that it aligns with the document data.

Table 10b MapText Field Types (Continued)

Field type	Description	Map the field type to this data element
Company	Specifies the recipient company name. Corresponds to the {{company}} FCL code.	Map this field to the name of the company that should receive the document. This information may be placed on the cover sheet or in a notification message.
Contact	Specifies the recipient name. Corresponds to the {{contact}} FCL code.	Map this field to the name of the person who should receive the document. This information may be placed on the cover sheet or in a notification message.
Cover	Specifies the cover sheet. Corresponds to the {{cover}} FCL code.	Map this field to the cover sheet file name. The cover sheet file must be stored in the directory RightFax\Production\Covers. If the file is not found in the directory when the document is created, the default cover sheet is used.

Table 10b MapText Field Types (Continued)

Field type	Description	Map the field type to this data element
Email	Specifies the e-mail address to send. <ul style="list-style-type: none"> • Notifications about the sent document to the host application. • The document via e-mail using the InternetLink Module. Corresponds to the {{email}} FCL code. For more information, see the <i>InternetLink Module Guide</i> .	The e-mail address of the person or company who is sending this document (used to send a notification to the e-mail address). The e-mail address of the person or company to which you are sending this document (requires InternetLink).
Fax	Specifies the fax number. Corresponds to the {{fax}} FCL code.	Map this field to the fax number where the document should be sent.

Table 10b MapText Field Types (Continued)

Field type	Description	Map the field type to this data element
Graphic	File name of a graphic.	<p>Map this field to the graphic file name. The file name must match the text in the mapped field. For example, Mark Jones' signature file must be named MarkJones.tif.</p> <p>The graphic file format must be .tif. The file must be stored in the directory RightFax\Production\Fors. If the file is not found in the directory when the document is created, then the image will not appear in the document.</p> <p>Example An image such as a signature or a company logo can be inserted.</p>
Include	<p>Specifies the file name of an include file.</p> <p>Corresponds to the {{include}} FCL code.</p> <p>An "include" file can contain commands and data that are common to many documents.</p>	<p>Map this field to the include file name.</p> <p>The file must be stored in the directory RightFax\Production\Include. If the file is not found in the directory when the document is created, then an error will occur and the document will not be processed.</p>

Table 10b MapText Field Types (Continued)

Field type	Description	Map the field type to this data element
Lookup Lookup2 - Lookup9	<p>Specifies the file name of a lookup table.</p> <p>Corresponds to the {{lookup}} FCL code.</p> <p>A lookup table can provide information that is not contained in the document that is sent from the host application, such as the recipient company name and fax number.</p>	<p>The text in the mapped field of the document will be compared to the first column in the specified lookup table. When a match is found, the associated data in the row will be used to send the document, included on the cover sheet, or included in a notification message.</p> <p>When you create a Lookup field, you can create or edit a lookup file. To do this, click Edit lookup file.</p> <p>The lookup table must be stored in the RightFax\Production\Include directory.</p> <p>Example The mapped field may contain a vendor ID. In the lookup table, vendor ID data includes the company name and fax number. The company name will be printed on the cover sheet, and the fax number will be used to send the document.</p>

Table 10b MapText Field Types (Continued)

Field type	Description	Map the field type to this data element
Owner	Specifies the sender's name. Corresponds to the {{owner}} FCL code.	Map this field to the sender's name. The name can appear on the cover sheet or in notification messages.
Page	Starts a new document when the number 1 appears in this field	Map this field to the page number. See also the Comment field type. The Page field type overrides the Comment field type.
SendFax	Use this field to print a document rather than fax it or send it through another method. The document will be printed if N, a space, or null characters are found in this field.	Map this field to the fax number. This field type is used when not all recipients have a fax number or other address for transmission. Such documents would be printed so that they could be mailed.

Table 10b MapText Field Types (Continued)

Field type	Description	Map the field type to this data element
Templatel D	Specifies the filter template to use to format the document.	Map this field to text that describes the document or indicates the document type. For example, the words "Purchase Order" may indicate that the Purchase Order template be used to format the document. When text in the mapped field matches the text specified, the specified template will be used to format the document.
User1 User 2 User3	Specifies user-defined information. Corresponds to the user-defined {{user1}}, {{user2}}, and {{user3}} FCL commands.	Map this field to text that you want to appear on the cover sheet or in a notification message.
Voice	Specifies the telephone number for the recipient.	Map this field to the phone number of the person who will be receiving the document. The number can appear on the cover sheet or in notification messages.

Troubleshooting

This section provides troubleshooting tips for building filter templates.

The background form is not aligned with data

In previewing a document that combines a background form with document data, the data does not align with the background form.

Table 10c Background Form Does Not Align with Data

Possible cause	Solution
The margins for the data are not set correctly. This causes the data to be too far from or too close to the left or top of the page.	Add or edit the FCL commands for the left and top margins. For instructions, see “Step 4: Insert FCL” on page 57 .
The fonts are not set correctly. This causes the data to be aligned correctly in some places but not in others.	Add or edit the FCL commands for the font. For instructions, see “Step 4: Insert FCL” on page 57 .
The number of lines on each page is not correct. This can cause the first page to align correctly, but the data on the second page is one line too low or high, the data on the third page is two lines too low or high, and so on.	Set the correct number of lines for each page of the document. For instructions, see “Step 3: Establish page length, orientation, and background forms” on page 57 .
The sample data from your host application may include print control language (PCL) in the first line. If so, the data on the first page will appear one line below the top margin, and the other pages will be aligned correctly.	Remove the PCL from the data stream that is sent from the host application. For instructions, see “Preparing to Create Filter Templates” on page 51 .

Document data does not match the background form

After correctly formatting documents for a period of time, the filter template unexpectedly does not match the document data. For example the background form does not line up correctly with the data.

Possible cause

Data from the host application may have changed, and the filter template must be updated. For example, a carriage return or a new data element may be added.

Solution

Review the data that is sent from the host application and the filter. If the data has changed, then you must change the filter template to accommodate it. For instructions, see [“Step 1: Overlay the sample data on a MapText document”](#) on [page 52](#).

Extra files are not included

A cover page, signature, background form, or other attached file is missing or is incorrect.

Possible cause

The file name or path name for the attachment is not correct.

Solution

Verify that the file name or path to the file is correct and that the file is stored in the correct directory. For instructions, see [“Preparing to Create Filter Templates”](#) on [page 51](#).

Documents are not addressed or are not sent correctly

Documents are incompletely addressed, the destination fax number is incomplete, or other information appears truncated in the document.

Possible cause

The fields for the data elements are not long enough.

Solution

Re-size the fields.

Creating an Input Channel for Filter Documents

You can create up to 70 unique input channels for the documents that must be processed with filter templates. One input channel can process documents for one filter template. The most common input channels are:

- Named pipe capture ([page 27](#))
- Directory scanning ([page 28](#))
- Serial capture ([page 29](#))
- TCP/IP port capture ([page 31](#))

The command line for the input channel must specify the filter template. The following table describes the command line options to use.

Table 10d Add Input Device Settings for a Named Pipe

Command line option	Description
-c "filter makedoc"	The option -c specifies one or more commands to execute on files received via the channel. <ul style="list-style-type: none"> • Filter.exe extracts the document data using the filter template. • Makedoc.exe begins the process of converting document data from the host application into a fax. The pipe symbol () separates the two commands in the command line. The commands must be enclosed in quotation marks.
-i <i>filename.mtd</i>	The option -i specifies that the specified MapText document (<i>filename.mtd</i>) should be included with the incoming data.
-f	Identifies <i>filename.mtd</i> as a file.

Example nplisten.exe -c "filter -i invoice.mtd -f \$\$ | makedoc" -p hplpr

In this example, Nplisten.exe will run the "filter" command against the data file (represented by \$\$) received on the named pipe. Filter will add FCL commands as designated by the invoice.mtd template and pipe the result to "makedoc" for processing.

■ ■ ■

Chapter 11

Specifying the Destination of the Document

A document can be sent to a fax number, an e-mail address, an SMS number, a printer, or a file. Use the following FCL commands to specify the destination for documents.

Table 11a Destination FCL Commands

Code	Description
<code>{{fax}}</code>	Sends documents to a fax number. If you have the InternetLink Module, then you can use an e-mail address with the <code>{{fax}}</code> command and send documents via e-mail. For more information, see the <i>RightFax InternetLink Module Guide</i> . The <code>{{altfax}}</code> command provides an alternative fax number that is used if the number provided with the <code>{{fax}}</code> command fails.
<code>{{type certified}}</code>	Sends the document as a certified e-mail message. This option requires the RightFax SecureDocs Module.
<code>{{type email}}</code>	Sends the document as the body of an e-mail message. This option requires the RightFax InternetLink Module.

Table 11a Destination FCL Commands (Continued)

Code	Description
<code>{{type file}}</code> <code>{{file}}</code> <code>{{list}}</code>	Sends the documents as files to a specified folder.
<code>{{type mime}}</code>	Sends the document as an attachment to an e-mail message. This option requires the RightFax InternetLink Module.
<code>{{type print}}</code>	Sends documents to a specified printer.
<code>{{type SMS}}</code> <code>{{smsmsg MessageText}}</code> <code>{{sms PhoneNum}}</code>	Specifies that a message will be sent to an SMS-compatible device.

Sending One Document to a Single Fax Number

To address a document to a single recipient, add the `{{fax}}` command to the FCL. The following example shows an FCL document that is addressed to 503-555-4489.

```
{{begin}}
{{fax 503-555-4489}}
Body of the document to be sent.
{{end}}
```

You can insert the `{{altfax}}` command as a backup fax number if the primary number fails:

```

{{begin}}
{{fax 503-555-4489}}
{{altfax 503-555-5240}}
Body of the document to be sent.
{{end}}

```

If you have purchased the RightFax InternetLink Module, you have more backup options. With the InternetLink Module you can specify a recipient fax number and e-mail address. If the fax number fails, the Integration Module can e-mail the document and notify you of the alternate sending method. For more information on this option and others provided by the InternetLink Module, see the *RightFax InternetLink Module Guide*.

For detailed information on the `{{fax}}` and `{{altfax}}` commands, including syntax and examples, see [Appendix B, "FCL Commands"](#).

Sending One Document to Many Recipients

Sending the same document to more than one recipient is called *broadcasting*. Broadcasting with FCL is accomplished by including additional `{{fax}}` commands just before the end of the FCL document. Captaris recommends that each `{{fax}}` command be followed by a `{{company}}` command for tracking purposes. Each broadcast document is issued its own document number and is treated as a separate entity for tracking and retry purposes.



Note Some restrictions exist for using cover sheets when you send documents to multiple recipients. For more information, see ["Using cover sheets with broadcasts"](#) on [page 78](#).

When the Integration Module encounters a second fax number in a document, it determines that the document is complete, writes the current page to the disk and schedules it to be sent to the preceding fax number. This process continues for each subsequent

fax number, and the Integration Module makes links between documents to keep disk usage at a minimum. An `{{end}}` command completes the loop for the final fax number.

The following example shows a single-page document to be broadcast to three recipients.

```

{{begin}} Dear customer,
Our office will be closed on Friday, March 25 for inventory. We will
be open again as usual on Monday.
Thanks,
International Finance Corporation
{{fax 555-3374}} {{company Salem Publishing Co.}} {{contact Juan
Garcia}} {{fax 206-555-6530}} {{company Oswego
Boats}} {{contact Gabriella Navarro}} {{fax
503-555-0905}} {{company West Coast Shippers}} {{contact Andy
Wells}}
{{end}}

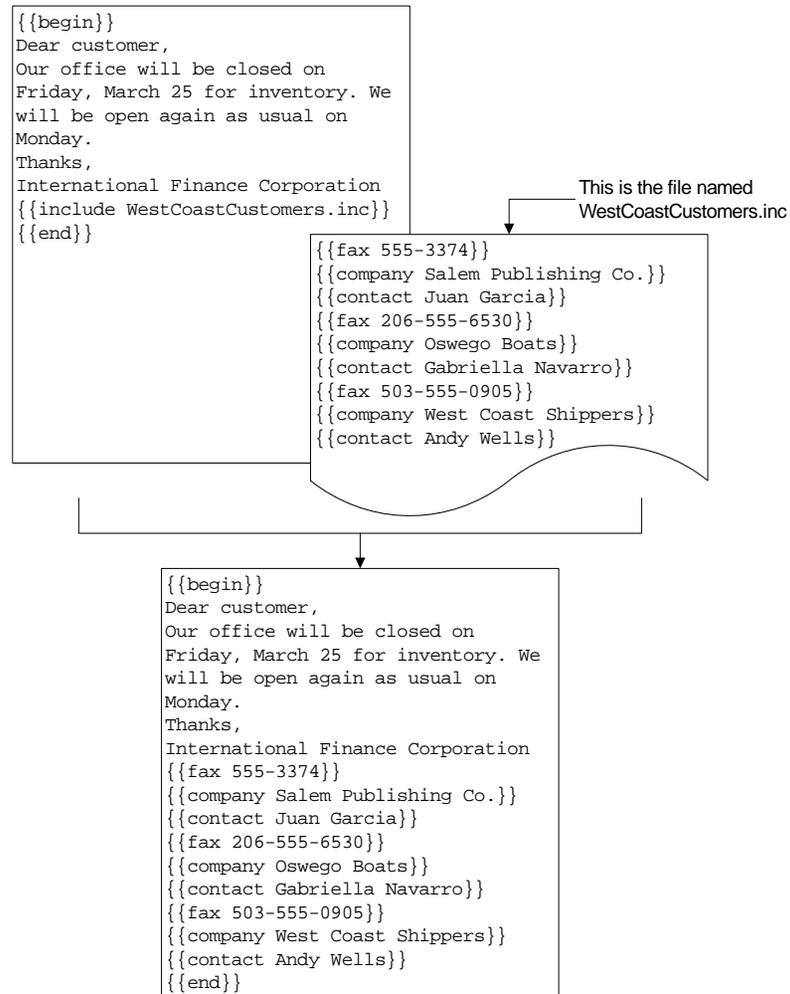
```

Sending One Document to Many Recipients With an Include File

A list of fax numbers can be associated with a document by using an *include file*. An include file can contain a list of FCL commands or data, and it is linked to a document with the `{{include}}` command.

“Figure 11.1: Broadcast Documents with an `{{Include}}` Command” shows the same broadcast document with an `{{include}}` command that inserts multiple addresses, instead of the addresses themselves.

Figure 11.1 Broadcast Documents with an `{{Include}}` Command



In “Figure 11.1: Broadcast Documents with an `{{Include}}` Command”, the data from `WestCoastCustomers.inc` would replace the `{{include}}` command in the original data. Thus, you need to insert only one command to address a document to multiple recipients. By having multiple broadcast include files stored on the Integration Module, you can easily broadcast a document to whatever group of numbers you choose.



Note Include files can do much more than store broadcast data. For information on different uses, how to create them, and where to save them, see Chapter 13, “Including the Same Files and Commands in Many Documents”. The following section deals with broadcast lists, just one form of include files.

Creating Lists of Recipients for a Broadcast

Use the following guidelines to create a list of recipients for a broadcast transmission.

Place the `{{fax}}` command first in each command on each line in an FCL document.

Correct `{{fax 503-555-3829}} {{company Acura of Salem}}`

Incorrect `{{company Acura of Salem}} {{fax 503-555-3829}}`

Do not insert any characters, such as spaces, between FCL commands that are listed in a line.

Correct `{{fax 503-555-3829}} {{company Acura of Salem}}`

Incorrect `{{fax 503-555-3829}} {{company Acura of Salem}}`

To create broadcast lists from a mainframe, server, or other host application

1. Include the `{{list filename}}` command in any document that contains a potential broadcast list.

For examples and usage information on this command, see [Appendix B, "FCL Commands"](#).



Note The `{{file}}` command has capabilities that are similar to the `{{list}}` command, although `{{list}}` is preferable for creating broadcast lists because it removes all white space, blank lines, etc. from the input file, while `{{file}}` does not. To determine which command will work best for your needs, you should compare the two. See [Appendix B, "FCL Commands"](#).

2. Send the document from the host application to the Integration Module.

The Integration Module creates a file (specified with the *filename* variable) in the RightFax\Production\Include folder. The content of this file is the data between `{{list filename}}` and `{{end}}` FCL codes.

Example If you were to send the following document to the Integration Module, it would create a file called Dealers.inc in RightFax\Production\Include.

```

{{begin}}
{{list dealers.inc}}
{{fax 503-555-0912}}
{{company ABSOLUTE
WHOLESALE}}
{{fax 503-555-7609}}
{{company ACURA OF SALEM}}
{{fax 555-8985}}
{{company ACURA OF PORTLAND}}
{{fax 541-555-8160}}
{{company ACURA OF
EUGENE}}
{{fax 503-555-4911}}
{{company HUTCHISON CO.}}
{{end}}

```

Dealers.inc would contain the following information:

```

{{fax 503-555-0912}}
{{company ABSOLUTE
WHOLESALE}}
{{fax 503-555-7609}}
{{company ACURA OF SALEM}}
{{fax 555-8985}}
{{company ACURA OF PORTLAND}}
{{fax 541-555-8160}}
{{company ACURA OF
EUGENE}}
{{fax 503-555-4911}}
{{company HUTCHISON CO.}}

```

To change broadcast lists created from a mainframe, server, or other host

1. Change the original document, as necessary.
2. Send the document, including the `{{list filename}}` command, from the host application to the Integration Module.

The existing file will be over-written with the new information, unless you change the name of the specified file.

To creating broadcast lists from a terminal, client, or other desktop computer

1. Using a text editor, word processor, spreadsheet, or database program that can produce plain text, type the required FCL commands in a text file.
2. Save the file with an .inc extension to the RightFax\Production\Include folder on the Integration Module.



Warning If you do not change the extension of the file to .inc when you store it in the \Include folder, errors will occur.

Using Cover Sheets in a Broadcast

Cover sheets in the Integration Module are known as “production” cover sheets. They are saved as .cov files. Production cover sheets cannot be used when broadcasting.

To broadcast a fax with a cover sheet, use a .pcl or .doc cover sheet and keywords. For information and instructions on .pcl and .doc cover sheets, see the *RightFax Administrator's Guide*.

If you are an upgrade customer who is familiar only with FCL commands, a list of equivalents is provided in the following table.

Table 11b Cover Sheet Keyword Equivalents

.cov keyword	.pcl or .doc keyword	Notes
billing	billinfo1	15 characters maximum.
billing2	billinfo2	15 characters maximum.
comment	to_citystate	59 characters maximum.
company	to_company	59 characters maximum.
contact	to_name	59 characters maximum.

Table 11b Cover Sheet Keyword Equivalents (Continued)

.cov keyword	.pcl or .doc keyword	Notes
coverttext	notetext	21 lines, each with each containing 69 characters maximum.
csi	from_phone	31 characters maximum.
fax	to_faxnum	31 characters maximum. Can contain alphabetical characters for macros.
onerror delete	faxflag_autodeleteall	Deletes successful documents.
onsuccess delete	faxflag_autodelete	
owner	from_name	59 characters maximum.
priority	ucPriority	Can be low (0), medium or normal (1), or high (2) priority. The default is low.
quality	finemode	
voice	to_contactnum	31 characters maximum.
winsecid	owner_id	The RightFax user ID of the originator of the fax.

The cover sheet is not cleared between broadcast fax numbers, so one {{cover}} command carries forward from document to document. You can turn on or off individual cover sheets inside the broadcast by using the {{cover none}} command.

Sending Documents to a Printer or File with FCL

Sending documents to a printer

To send a document to a printer, add the `{{type print}}` command to the FCL. The document will print at the default printer specified in the Integration Module Configuration program. You can change the printer with the `{{lp}}` command. The following example shows an FCL document that will be printed.

```

{{begin}}
{{type print}}
Body of the document to be sent.
{{end}}

```

Sending documents to a file

The Integration Module can print documents to two different kinds of files: plain text (ASCII) files or .tif files. The `{{type file}}` command creates a .tif image of the host document and stores it in a folder that you specify. In the following example, the file MyDocument.tif would be saved in the RightFax\Production folder.

```

{{begin}}
{{type file RightFax\Production\MyDocument}}
Body of the document to be sent.
{{end}}

```

The `{{file}}` and `{{list}}` commands creates a plain text (ASCII) file with an .inc extension in the RightFax\Production\Include folder. `{{List}}` strips all leading white space and blank lines, but `{{file}}` does not. In the following example, the file MyDocument.inc would be saved in the RightFax\Production\Include folder.

```

{{begin}}
{{file MyDocument}}
Body of the document to be sent.
{{end}}

```

You can also specify a full path to another folder:

```

{{begin}}
{{list c:\Marketing Files\Reports\MyDocument}}
Body of the document to be sent.
{{end}}

```

For more information on the `{{file}}` and `{{list}}` commands, including syntax and examples see [Appendix B, "FCL Commands"](#).

■ ■ ■

Chapter 12

Creating and Linking Background Forms

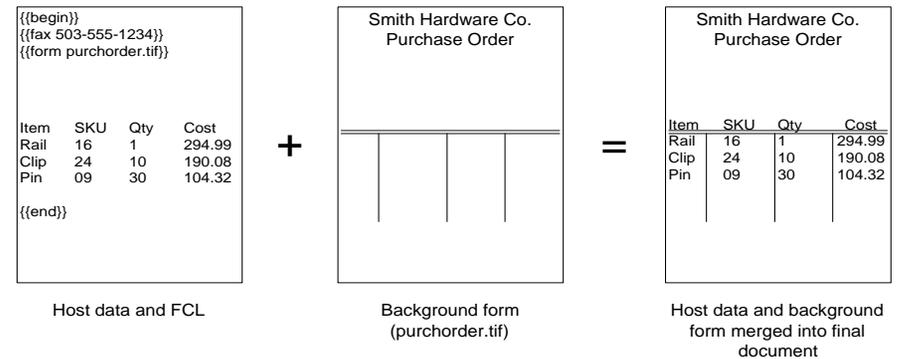
This chapter explains background forms—the graphic files that are merged with a document from the host application to create a finished document that looks like it was printed on a preprinted form. Linking a background form to a host document is different from inserting or attaching a graphic file.

Understanding Background Forms

Typically, background forms are used to replace documents that would normally be printed on preprinted forms and then mailed or manually faxed. The Integration Module merges an electronic background form with the host data and then sends the finished document per your specifications.

The following figure shows a simple example of how the Integration Module merges the data from the host application with a background form.

Figure 12.1 Host data Merged with a Background Form



Creating Background Forms

You can create background forms by scanning an existing paper form, or you can use a drawing application to create a form.

All background forms must be in class F, group 3 or 4 TIF format.

Creating forms with a drawing program

You can use any application on any platform to create a background form. The file must be class F, group 3 or 4 TIF, which is the standard format for faxes.

Scanning paper forms

If you have a paper form that you want to duplicate, scan the form at 200 dpi.

Storing background forms

Save background forms with any descriptive name as a class F, group 3 or 4 TIF file in the RightFax\Production\Forms folder.

For detailed information on the `{{form}}` command, including syntax and examples see [Appendix B, "FCL Commands"](#).

■ ■ ■

Linking Background Forms

Use the `{{form}}` FCL command to link a background form to a document.

Example

```
{{begin}}
{{fax 503-555-9023}}
{{form invoice.tif}}
Host data.
{{end}}
```

This example would merge the host data with a background form called `invoice.tif`, which resides in the RightFax\Production\Forms folder.

In this example, the `{{form invoice.tif}}` command could also be `{{form invoice}}` (without the `.tif` extension). The extension `.tif` is the default.

Chapter 13

Including the Same Files and Commands in Many Documents

To include the same files and commands in many documents, you can create an *include file*. An include file contains information that will be duplicated in multiple documents. For example, an include file can contain a list of fax numbers, so that the same documents can be addressed and sent to multiple recipients.

Understanding Include Files

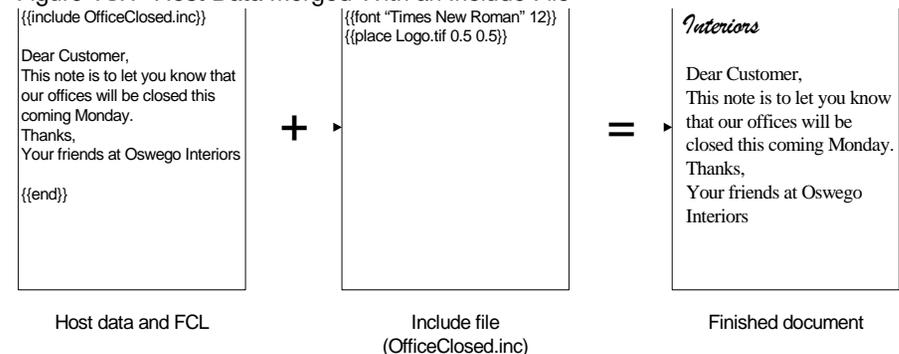
You have two options for creating and using include files:

- Global include files are provided in the RightFax\Production\Include folder when you install the Integration Module software. The files are empty. By default, global include files are linked to all documents but have no function unless you put information into them. Typically, global include files are used to establish default attributes for all documents.
- Standard include files. You can create standard include files and link them to documents with the `{{include}}` FCL command. Typically, standard include files are used to change an attribute in a large batch of documents, for example, to override a default attribute for a group of documents.

Example You can use a global include file to set the default font for all documents to Arial 11-point. To send a batch of documents using a different font, use a standard include file to change the font to Times 12-point.

The following example shows how a standard include file is used to create and send a document.

Figure 13.1 Host Data Merged With an Include File



In this example, the include file specifies a top margin (`{{tm}}`), a left margin (`{{lm}}`), and a font. It also specifies that the file Logo.tif be placed at specific coordinates.

Understanding the content of include files

You can place FCL commands in include files to reduce the programming necessary to place commands in documents. You can use any FCL command.

Also, you can place sentences, paragraphs, entire letters, and even multi-page documents in include files. Because include files are plain text (ASCII) documents, you must format the text with FCL commands. Any text will appear in the body of the finished document at the point where the `{{include}}` command or global include file appears.

Understanding the placement of global include files

If the global include files contain content, the content will be inserted at default locations in all documents, as described in the following table.

Table 13a Placement of Data from Global Include Files

File name	Default location in the FCL documents
Global.beg	Beginning of document.
Global.end	End of document.
Global.def	Immediately after every fax number.
Global.inc	Immediately before every fax number and at the end (before global.end).

The following table illustrates the default placement of information from these global include files.

Table 13b Example FCL Document with Global Include Files

FCL document	Placement of data
<code>{{BEGIN}}</code>	<code>{{BEGIN}}</code> Global.beg
<code>{{Company Test}}</code>	<code>{{Company Test}}</code>
<code>{{FAX 968-9601}}</code>	<code>{{FAX 968-9601}}</code> Global.def
<code>{{Contact Jay Doe}}</code>	<code>{{Contact Jay Doe}}</code>
<code>{{FAX 968-9602}}</code>	Global.inc <code>{{FAX 968-9602}}</code> Global.def
<code>{{Contact Sid Brea}}</code>	<code>{{Contact Sid Brea}}</code>
<i>Body of the document to be sent.</i>	<i>Body of the document to be sent.</i>
<code>{{END}}</code>	Global.inc Global.end <code>{{end}}</code>

Global.beg and global.end are used for general FCL commands that are used only once per FCL document. Usually, these are FCL commands that are carried over from one document to the next when broadcasting, such as `{{winsecid}}`. `{{Winsecid}}` is not reset automatically in broadcast documents, so placing it in Global.beg causes it to become a default, setting a RightFax mailbox to be used by all production documents.

Global.def and Global.inc are more commonly used than Global.beg and Global.end.

Because of its placement in the FCL (after every fax number), Global.def can be overridden if there are conflicting FCL commands in the data sent from the host.

Example If you insert `{{priority low}}` in Global.def, all documents will be sent at that priority except when a different `{{priority}}` command is included in the FCL coming from the host for that particular recipient.

Because of its placement in the FCL (before every fax number and at the end, before Global.end), Global.inc overrides any conflicting FCL.

Example If you insert `{{priority low}}` in Global.inc, all documents will use that priority, ignoring any conflicting `{{priority}}` commands that are included in the FCL coming from the host.

Creating Include Files

Creating standard include files

You can use any word processor that will produce plain text (ASCII), such as Microsoft Notepad, to create an include file. Insert any FCL or other text that you want to appear in multiple documents. Save the include file with an .inc extension in the RightFax\Production\Include folder.

Inserting content into global include files

To open any of the global include files in the RightFax\Production\Include folder, double-click the file. In the **Open With** dialog box, select **Notepad**.

Insert any FCL or other text that you want to appear in multiple documents.



Warning Putting incorrect FCL into a global include file can cause system-wide errors. For example, if you put `{{include file.inc}}` in an include file called `File.inc`, then the file will attempt to insert itself in the FCL document. The resulting infinite loop will cause RightFax to stop responding.

Every document sent through the Integration Module will use the information that you add to a global include file. You must understand the function of any FCL you insert in these files, including the different effects of a given command depending on where it appears in a document. (These effects will dictate into which global include file you should insert certain commands.)

Storing include files

Save include files with any descriptive name with an `.inc` extension in the RightFax\Production\Include folder.

Linking Include Files

Linking standard include files with the `{{include}}` command

Use the `{{include}}` command to link an include file (one that you have created and saved with an `.inc` extension in the RightFax\Production\Include folder) with a document.

```
{{begin}}
{{fax 503-555-9023}}
{{include OfficeClosed.inc}}
Host data.
{{end}}
```

This example would replace the `{{include}}` command with the content of the file called `OfficeClosed.inc`, which resides in the RightFax\Production\Include folder.



Note In this example, the `{{include OfficeClosed.inc}}` command could also be `{{include OfficeClosed}}` (without the `.inc` extension). The `.inc` extension is assumed when an include file is identified as part of an `{{include}}` command. Thus, you can insert it or omit it. Accordingly, if you create and attempt to link an include file that is not saved with an `.inc` extension, then errors will occur.

For detailed information on the `{{include}}` command, including syntax and examples, see [Chapter 9 “Setting Defaults for FCL Documents”](#).

Linking global include files

The global include files are linked by default to every document that the Integration Module processes. You do not need to link the global include files to documents.

Because every document is linked to all the global include files, you must use caution when inserting commands and other text into a global include file. Inserting incorrect FCL into a global include file can cause system-wide errors.

■ ■ ■

Chapter 14

Creating and Attaching Cover Sheets

A cover sheet is the first page of a faxed document. It usually includes information about the fax, such as the recipient's name and fax number, the sender's name and telephone number, and the total number of pages in the fax. RightFax cover sheets are templates with placeholders for the information. The placeholders are filled with data and attached to a document when it is processed for faxing.

Attaching a cover sheet to a document is different from attaching and embedding files. For more information on attaching other kinds of files, see [Chapter 19, "Attaching and Embedding Files, Signatures, and Graphics"](#).

The RightFax system supports two kinds of cover sheets:

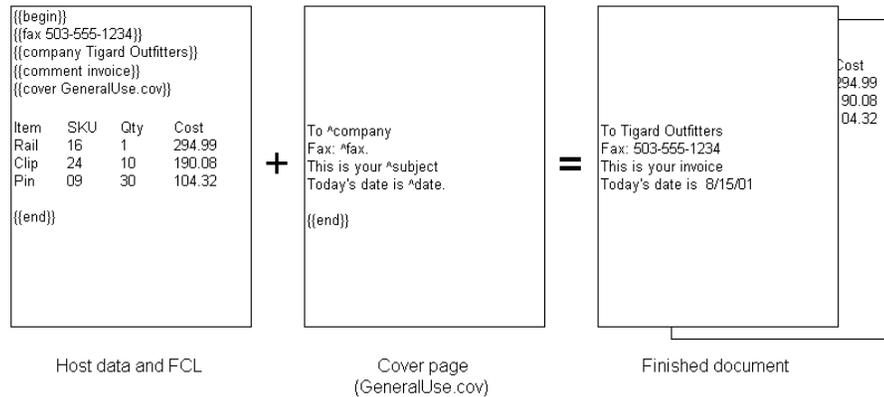
- *Production* cover sheets can contain FCL codes, text, and keywords. These cover sheets are attached to documents by inserting the `{{cover}}` command in the documents. The files are named with the extension `.cov`, and they are stored in the folder `RightFax\Production\Covers`. The Integration Module generates these cover sheets.
- *Enterprise* cover sheets can contain embedded codes, text, and keywords. These cover sheets are attached to documents by a user when the fax is created or by default settings for the user or the enterprise. The file format can be print control language (`.pcl`) or Microsoft Word (`.doc`), and they are stored in the folder `RightFax\FCS`. The RightFax server generates these cover sheets. You can use `.pcl` and `.doc` cover sheets with FCL documents. For details on `.pcl` and `.doc` cover sheets, see the *RightFax Administrator's Guide*.

This chapter describes production cover sheets.

Understanding Production Cover Sheets

The following example illustrates how the Integration Module generates a simple cover sheet and then attaches it as the first page of a fax.

Figure 14.1 Sample Cover Sheet



In this example, the command `{{cover GeneralUse.cov}}` retrieves the template file `GeneralUse.cov` from the `RightFax\Production\Covers` folder. `GeneralUse.cov` contains text and keywords, which the Integration Module populates with information from the document data from the host application.

Notice that “To ^company” in `GeneralUse.cov` becomes “To Tigard Outfitters” in the finished cover sheet. That is because the ^company keyword maps to the content of the `{{company}}` FCL command.

Using cover sheets with broadcasts

Production cover sheets cannot be used when broadcasting. For more information, [“Using Cover Sheets in a Broadcast” on page 69](#).

Creating Cover Sheet Templates

Using default cover sheet templates

The Integration Module includes a cover sheet called `Auto.cov`. This file is the default cover sheet, unless you specify a different one with the Integration Module Configuration program. For information on establishing a default cover sheet, [“Setting General Defaults” on page 45](#).

`Auto.cov` contains generic information such as the company, contact, owner, fax number, and total pages in the fax. It is located in the folder `RightFax\Production\Covers`.

If the `{{cover}}` command is included in the document data, but a cover sheet name is not specified, then `Auto.cov` will be used.

Creating a basic cover sheet template

1. Open Microsoft Notepad or another word processing application that will produce plain text.
2. Type text, FCL, or keywords in the template.
 - To extract blocks of text from the document data to the cover sheet, [“Adding blocks of text to a cover sheet from the document data” on page 79](#).
 - To use FCL codes, [“FCL for cover sheets” on page 81](#).
 - To use keywords, [“Keywords for cover sheets” on page 79](#).
3. Type `{{end}}` in the last line of the cover sheet.
4. Save the file with the extension `.cov` in the folder `RightFax\Production\Covers`.

Adding blocks of text to a cover sheet from the document data

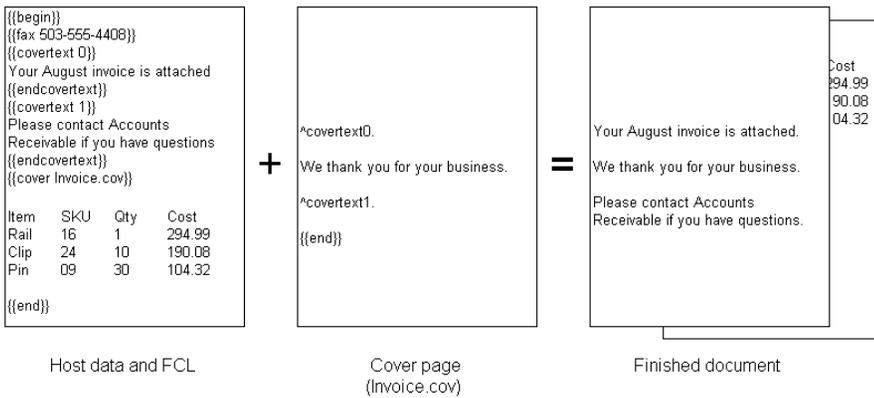
In addition to typing text in a cover sheet template, you can use text from the document data to populate a cover sheet. This is useful when a block of text on a cover sheet must be unique in each document.

Use the `{{covertext}}` and `{{endcovertext}}` commands in the document data and the `^covertext` keyword in the cover sheet template. The following figure shows the relationship between the `{{covertext}}` command and the `^covertext` keyword.



Note Coversheets provide a maximum of 75 characters, per line.

Figure 14.2 How the Integration Module Uses the `{{Covertext}}` Command



For detailed information on the `{{covertext}}` and `{{endcovertext}}` commands see [Appendix B, “FCL Commands”](#).

For more information on the `^covertext` keyword, “[Keywords for cover sheets](#)” on [page 79](#).

Word-wrapping cover sheet note text

By default, text entered in the notes field of a cover sheet will wrap at 70 characters. Use the steps below to customize this setting.

1. On the RightFax server, open the Windows registry and browse to `HKLM\Software\RightFax\Production`.
2. Create a new DWORD value called `CoverSheetWordWrapMax`.
3. Edit this value to contain a decimal number that represents the maximum number of characters per line. When finished, click OK.
4. Close the Windows registry.

Keywords for cover sheets

The following table lists keywords you can use in production cover sheets. Each keyword is populated with the content from the FCL code that is included with the document data. In the cover sheet template, each keyword must begin with the carat symbol (^).

For information on the FCL commands, see [Appendix B, “FCL Commands”](#).

Table 14a Cover Sheet Keywords and FCL Equivalents

Keyword	FCL equivalent	Notes
<code>^altfax</code>	<code>{{altfax}}</code>	
<code>^billing</code>	<code>{{billing}}</code>	
<code>^billing2</code>	<code>{{billing2}}</code>	
<code>^comment</code>	<code>{{comment}}</code>	
<code>^company</code>	<code>{{company}}</code>	
<code>^contact</code>	<code>{{contact}}</code>	

Table 14a Cover Sheet Keywords and FCL Equivalents (Continued)

Keyword	FCL equivalent	Notes
<code>^covertext n</code>	<code>{{covertext}}</code> <code>{{endcovertext}}</code>	Text specified by the <code>{{covertext}}</code> and <code>{{endcovertext}}</code> commands in the document data. Multiple blocks of text can be used on a single cover sheet. Specify the index number of the text block. Valid index numbers are 0-9, and the index number is not optional. If only a single text block is to be used on a cover sheet, use 0 as the index number.
<code>^csi</code>	<code>{{csi}}</code>	If no caller subscriber information (csi) is specified, the <code>^csi</code> variable will use the default fax number set in the Integration Module Configuration program. Changes are global; you can change CSI on a per-document basis, but not on the banner line.
<code>^date</code>	<code>{{date}}</code>	If no <code>{{date}}</code> is specified, the current date will be used.
<code>^dept</code>	<code>{{dept}}</code>	
<code>^docnum</code>	N/A	The document number assigned by the RightFax Integration Module.
<code>^email</code>	<code>{{email}}</code>	This command can be used if you have licensed the InternetLink Module. The recipient's e-mail address.
<code>^emailcc</code>	<code>{{email}}</code>	This command can be used if you have licensed the InternetLink Module. The e-mail address for an additional recipient.

Table 14a Cover Sheet Keywords and FCL Equivalents (Continued)

Keyword	FCL equivalent	Notes
<code>^empid</code>	<code>{{empid}}</code>	
<code>^fax</code>	<code>{{fax}}</code>	
<code>^file</code>	<code>{{type file}}</code>	The file name for <code>{{type file}}</code> documents.
<code>^from</code>	<code>{{email}}</code> <code>{{from}}</code>	The <code>{{from}}</code> command can be used if you have licensed the InternetLink Module.
<code>^localfax</code>	N/A	Replaced with the local fax number set in the Integration Module Configuration program.
<code>^oecopies</code>	<code>{{onerror print}}</code>	When <code>{{onerror}}</code> action is set to print, <code>^oecopies</code> is replaced with the number of copies to be printed.
<code>^onerror</code>	<code>{{onerror}}</code>	Replaced with the <code>{{onerror}}</code> action specified for this document, or if none is specified, the default set in the Integration Module Configuration program.
<code>^onsuccess</code>	<code>{{onsuccess}}</code>	The <code>{{onsuccess}}</code> action specified for this document. If none is specified, the default set in the Integration Module Configuration program will be used.
<code>^owner</code>	<code>{{owner}}</code>	
<code>^pages</code>	N/A	The number of attachments in the fax. The number of pages in an attached file are not counted.
<code>^phone</code>	<code>{{fax}}</code>	

Table 14a Cover Sheet Keywords and FCL Equivalentents (Continued)

Keyword	FCL equivalent	Notes
^quality	{{quality}}	
^rti	{{rti}}	If no {{rti}} string is specified, the ^rti variable will use the default set in the Integration Module Configuration program.
^subject	{{comment}} {{subject}}	The {{subject}} command can be used if you have licensed the InternetLink Module.
^termid	{{termid}}	
^time	{{time}}	If no {{time}} is specified, the current time will be used.
^tranid	{{tranid}}	
^type	{{type}}	
^unique_id	{{unique_id}}	Maps to the unique_id field in the RightFax database.
^user1	{{user1}}	
^user2	{{user2}}	
^user3	{{user3}}	
^voice	{{voice}}	
^winsecid	{{winsecid}}	The RightFax user ID of the originator of the fax.

FCL for cover sheets

Several informational FCL commands are designed specifically for cover sheets.

The following table lists the commands and gives a brief explanation. For a comparison of the commands listed here and the keywords that you can use in cover sheets, see [“Table 14a: Cover Sheet Keywords and FCL Equivalentents”](#).

For more detailed information on each of the following commands, including syntax and examples see [Appendix B, “FCL Commands”](#).

Table 14b Cover Sheet FCL Commands

Command	Description
{{comment}}	Stores any user-defined message specific to the document. This command is sometimes also used to populate variables in notifications.
{{company}}	Stores a company name for the current document. This command is sometimes also used to populate variables in notifications.
{{contact}}	Stores the contact name for the current document. This command is sometimes also used to populate variables in notifications.
{{CSI}}	Associates called-subscriber information (CSI) with the document. This is usually the general fax number for the enterprise. You can set a default CSI in the Integration Module Configuration program. For more information, see Chapter 9, “Setting Defaults for FCL Documents” .
{{email}}	Stores the return sender’s return e-mail address of the document. This command is sometimes also used to populate variables in notifications.
{{empid}}	Specifies the employee ID of the fax owner. This command is most often used to populated variables in notifications, but is sometimes also used with cover sheets.
{{owner}}	Specifies the document owner’s name. This command is sometimes also used to populate variables in notifications.

Table 14b Cover Sheet FCL Commands (Continued)

Command	Description
{{replyto}}	Specifies a recipient for a notification. You can request that an HTTP post be sent back to the host as a notification when you use the RightFax XML Interface. ReplyTo is the field in the submit post that the XML Interface populates to determine where to send the notification.
{{RTI}}	Specifies the remote terminal identification from which the document originated. This is typically the sending company's name. You can set a default name for the sending company in the Integration Module Configuration program. For more information, see Chapter 9, "Setting Defaults for FCL Documents" .
{{termID}}	Specifies the terminal identification from which the document originated. This command is most often used to populated variables in notifications, but is sometimes also used with cover sheets.
{{tranID}}	Sets the identification of the transaction that produced the document. This command is most often used to populated variables in notifications, but is sometimes also used with cover sheets.
{{uniqueID}}	Sets an identification number for each destination (fax number) within the document. This command is used most often for tracking. The Integration Module will generate a UniqueID unless you specify one in your FCL. Then, you can track the document in FaxUtil based on the UniqueID. Secondarily, this command is sometimes used in cover sheets and with notifications.

Table 14b Cover Sheet FCL Commands (Continued)

Command	Description
{{user1}}, {{user2}}, {{user3}}	Include one or more of these commands in the host document so that the originator (person, group, etc.) of the document can be identified and correctly sent the notification.
{{userID}}	Identifies the creator of this document. This command is most often used to populated variables in notifications, but is sometimes also used with cover sheets.
{{voice}}	Sets the voice number to be associated with the document. This command is most often used to populated variables in notifications, but is sometimes also used with cover sheets.

Attaching a Cover Sheet to a Document with FCL

Use the {{cover}} command to attach a production cover sheet to a document.

```

{{begin}}
{{fax 503-555-9023}}
{{cover FaxCoverPage.cov}}
Host data.
{{end}}

```

In this example, the Integration Module would retrieve the file called FaxCoverPage.cov from the RightFax\Production\Covers folder and attach it as the first page of the document.

For detailed information on the {{cover}} command, see [Appendix B, “FCL Commands”](#).



Note In this example, the {{cover FaxCoverPage.cov}} command could also be {{cover FaxCoverPage}} (without the .cov extension). The .cov extension is assumed when a cover sheet is identified as part of a {{cover}} command. Thus, you can insert it or omit it. Accordingly, if you create and attempt to attach a production cover sheet that is not saved with a .cov extension, then errors will occur.

RightFax Cover Sheet Logic

Because you can set a default cover sheet in the Integration Module Configuration program, set a default cover sheet for a RightFax user, and specify a cover sheet with the {{cover}} FCL command, the following table will help clarify which cover sheet will be used in different scenarios.

Table 14c Cover Sheet Logic

With these configuration settings			This cover sheet will be used			
RightFax user default cover sheet	Integration Module default cover sheet	FCL in document	The specified file name	The user default	The Integ. Module default	No cover sheet
None set	“none”	“cover” (.pcl, .doc, or .cov)	X			
None set	“none”	“none”				X
None set	“none”	“rfdefault”				X
None set	“none”	No command				X
None set	Cover.cov	“cover” (.pcl, .doc, or .cov)	X			
None set	Cover.cov	“none”				X
None set	Cover.cov	“rfdefault”				X
None set	Cover.cov	No command			X	
Cover.pcl	“none”	“cover” (.pcl, .doc, or .cov)	X			
Cover.pcl	“none”	“none”				X
Cover.pcl	“none”	“rfdefault”		X		
Cover.pcl	“none”	No command				X
Cover.pcl	Cover.cov	“cover” (.pcl, .doc, or .cov)	X			
Cover.pcl	Cover.cov	“none”				X
Cover.pcl	Cover.cov	“rfdefault”		X		

Table 14c Cover Sheet Logic (Continued)

With these configuration settings			This cover sheet will be used			
RightFax user default cover sheet	Integration Module default cover sheet	FCL in document	The specified file name	The user default	The Integ. Module default	No cover sheet
Cover.pcl	Cover.cov	No command			X	
Cover.pcl	"rfdefault"	"cover" (.pcl, .doc, or .cov)	X			
Cover.pcl	"rfdefault"	"none"				X
Cover.pcl	"rfdefault"	"rfdefault"		X		
Cover.pcl	"rfdefault"	No command		X		

Example Cover Sheet and Document

Cover sheet FCL and text

```

{{place logo.tif 5 1}}
{{lm 0.5}}
{{moveto .5 1.5}}Palatine Hill
Association
0615 S.W. Palatine Hill Road
Portland, Oregon 97219
{{lm 1.0}}
{{moveto 1 3.5}}FAX TRANSMISSION

Date:      ^date
Page:      1 of ^pages

To:        ^contact
Company:   ^company
Date:      ^date

From:      ^owner
Company:   Palatine Hill Association
Fax:       ^csi

Subject:   ^comment

Notes:

^covertext 0
{{end}}
    
```

Host document data that contains FCL

```

{{begin}}
{{cover faxcover.cov}}
{{owner Adam Guenther}}
{{comment Purchase Order}}
{{covertext 0}}
Let me know if you have questions!
{{endcovertext}}
{{moveto 0 .5}}
Purchase Order
{{linewidth 4}}
{{lineto 8.5,y}}
{{lm 1.0}}
ITEM PRICE QUANTITY TOTAL
11023 $21.00 2 $42.00
=====
$42.00
{{company Troutdale Imports}}
{{contact Jack Dundee}}
{{fax 555-465-0987}}
{{end}}
    
```

Finished cover sheet and document

Figure 14.3 The Cover Sheet is Attached as the First Page of the Document

Purchase Order			
ITEM	PRICE	QUANTITY	TOTAL
11023	\$21.00	2	\$42.00
			=====
			\$42.00



Palatine Hill Association
0615 S.W. Palatine Hill Road
Portland, Oregon 97219

FAX TRANSMISSION
Date: 08/31/01
Page: 1 of 2

To: Jack Dundee
Company: Troutdale Imports
Date: 08/31/01

From: Adam Guenther
Company: Palatine Hill Association
Fax: 503.968.9601

Subject: Purchase Order

Notes:
Let me know if you have questions!



Chapter 15

Creating Lines, Boxes, and Other Shapes

A series of FCL commands can be used to create graphical shapes to appear in documents. Creating and configuring these shapes can supplement or replace background forms or graphics.

For information on background forms, see [Chapter 12, “Creating and Linking Background Forms”](#). For information on graphics, see [Chapter 19, “Attaching and Embedding Files, Signatures, and Graphics”](#).

Understanding the Commands That Create Shapes

The following table lists the FCL commands for drawing shapes and provides a brief description. For detailed information on each command, including syntax and examples, see [Appendix B, “FCL Commands”](#).

Table 15a FCL Commands that Create Shapes

Command	Description
{{Box}}	Draws a box whose size and location at specified coordinates. You can also place text inside the box.
{{EndPoly}}	Ends a polygon that is started with the {{startpoly}} command.

Table 15a FCL Commands that Create Shapes (Continued)

Command	Description
{{FillBox}}	Same as the {{box}} command. The box can be filled with white or black. This is frequently used to hide information that cannot be removed easily from the data stream coming from host application. You cannot place text inside the box.
{{Line}}	Draws a line whose starting and ending points are specified by coordinates that you supply.
{{LineTo}}	Draws a line whose starting point is the current cursor location and whose ending point is specified by coordinates that you supply.
{{LineWidth}}	Lets you specify the width, in points, of the lines that you draw with other commands.
{{RBox}}	Same as {{box}}, but the coordinates for size and location are established relative to the current cursor location. In contrast, the coordinates for the {{box}} command are measured from the upper-left corner of the document.
{{RFillBox}}	Same as {{fillbox}}, but the coordinates for size and location are established relative to the current cursor location. In contrast, the coordinates for {{fillbox}} are measured from the upper-left corner of the document.

Table 15a FCL Commands that Create Shapes (Continued)

Command	Description
{{RLine}}	Same as {{line}}, but the coordinates that you supply to determine length and location are established relative to the current cursor location. In contrast, the coordinates you give with {{line}} are measured from the upper-left corner of the document.
{{RLineTo}}	Same as {{lineto}}, but the coordinates that you supply to determine length and location are established relative to the current cursor location. In contrast, the coordinates you give with {{line}} are measured from the upper-left corner of the document.
{{RStartPoly}}	Same as {{startpoly}}, but the coordinates that you supply to determine the starting point are established relative to the current cursor location. In contrast, the coordinates you give with {{startpoly}} are measured from the upper-left corner of the document.
{{StartPoly}}	Starts a polygon at a point that you specify by supplying coordinates. The polygon is completed with a series of {{lineto}} commands and finished with the {{endpoly}} command.

Using the Most Common Commands That Create Shapes

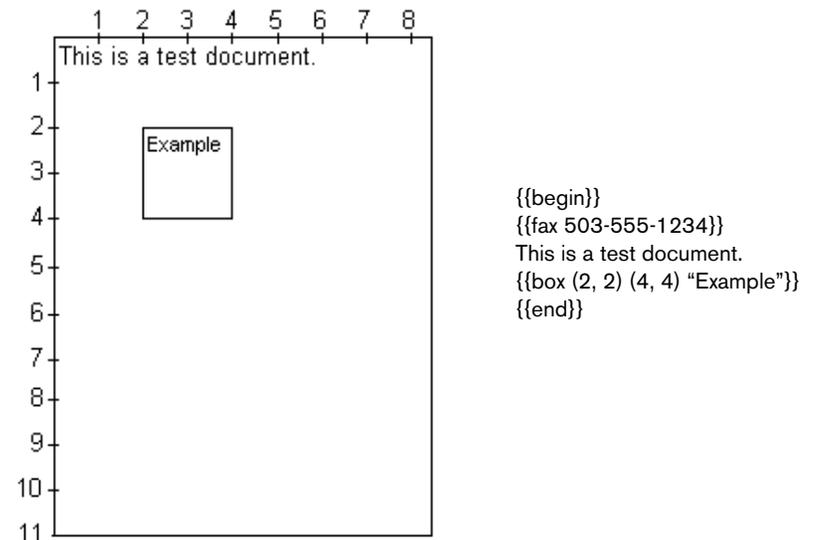
This section lists the most common commands and provides examples for creating shapes, such as a line or a box, with FCL.

Example FCL code and finished documents are provided. The sample documents have horizontal and vertical numbers at the top and left margins to indicate the grid for the x- and y-coordinates for the placement of the shapes. The numbers will not appear in the finished document.

Box

The box command creates a box at specified coordinates and places text in it. You supply coordinates for the upper-left and lower-right corners. In the following example, (2, 2) is the coordinate for the upper-left corner of the box, and (4, 4) is the coordinate for the lower-right corner of the box.

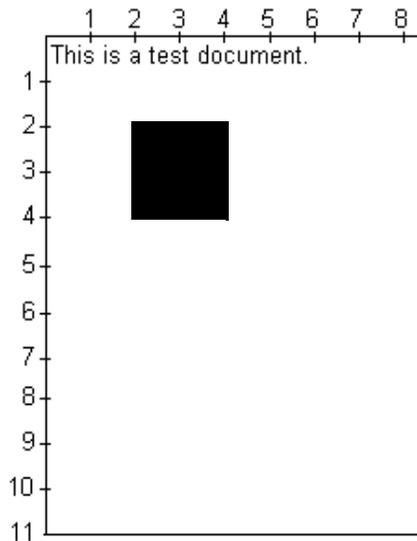
Figure 15.1 Box Command



Fillbox

The fillbox command is the same as `{{box}}`, except that you can fill it with black or white. You might use a white-filled box to cover part of a document that should not appear in the finished document.

Figure 15.2 Fillbox Command

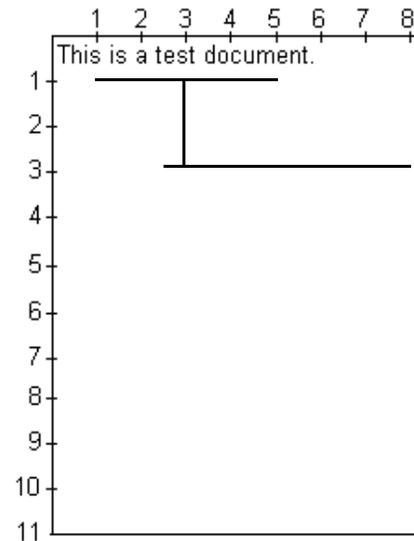


```
{{begin}}
{{fax 503-555-1234}}
This is a test document.
{{fillbox (2, 2) (4, 4) black}}
{{end}}
```

Line

The line command draws a line between specified coordinates. The following example shows three `{{line}}` commands. In the first, (1, 1) are the x- and y-coordinates for the beginning of the line, and (5, 1) are the coordinates for the end of the line.

Figure 15.3 Line Command



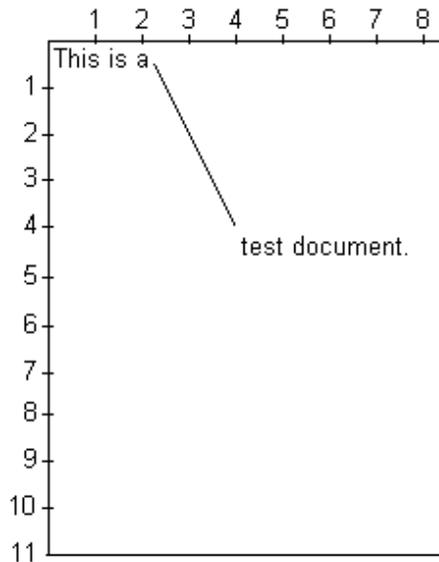
```
{{begin}}
{{fax 503-555-1234}}
This is a test document.
{{line (1, 1) (5, 1)}}
{{line (3, 1) (3, 3)}}
{{line (2.5, 3) (8, 3)}}
{{end}}
```

Lineto

The lineto command draws a line between the current cursor location and specified coordinates. The current cursor location is often where the FCL command appears in a document but not always. The two examples in this section show different options for this command.

The first example begins the line at the current location when the Integration Module finds the `{{lineto}}` command. In this case, the location is immediately after “This is a”. The line ends at the coordinates (4, 4). Then, the remainder of the text appears.

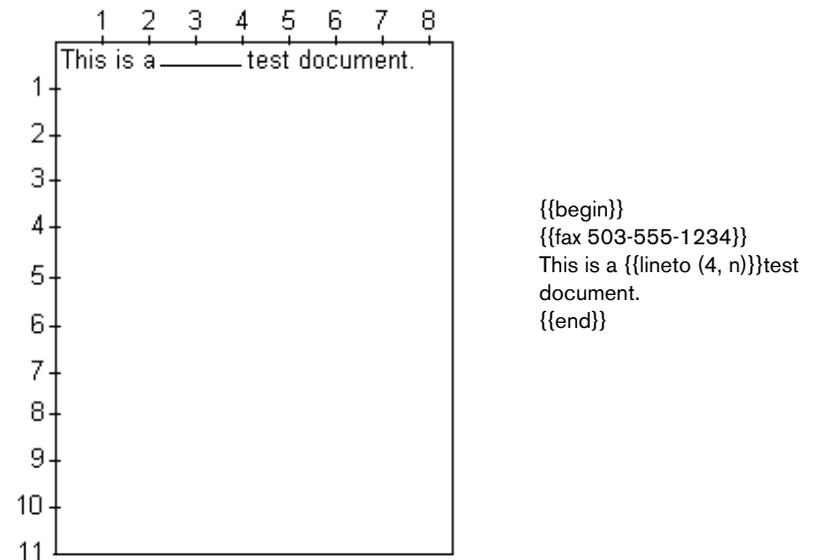
Figure 15.4 Lineto Command



The second example starts the line at the current location when the Integration Module finds the `{{lineto}}` command. In this case, the location is immediately after “This is a”. The line ends at the coordinates (4, y), which is 4 units to the right of the starting point,

and n units below the starting point (where n is the current position; thus, the position does not move down). Then, the remainder of the text appears.

Figure 15.5 Lineto Command



Rbox

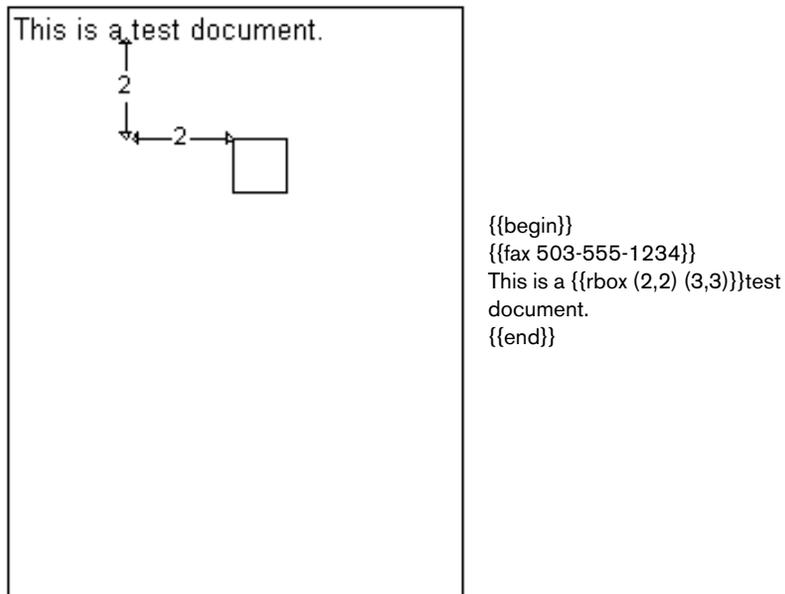
The `rbox` command draws a box with a starting point that is relative to the current cursor location. In contrast, the coordinates you specify with the `{{box}}` command are measured from the upper-left corner of the document.

The current cursor location is often where the `FCL` command appears in a document but not always. The two examples in this section show different options for this command.

The first example starts the box at (2, 2) from the current location when the Integration Module finds the `{{rbox}}` command. In this case, the location is immediately after “This is a”. The box ends at the coordinates (3, 3) from the current location. Then, the remainder of the text appears.

The following example does not show grid numbers, because the position of the box is relative to a point that can move from document to document. The arrows in the example document indicate that the box starts (2, 2) from the current cursor location. These arrows would not appear in the finished document.

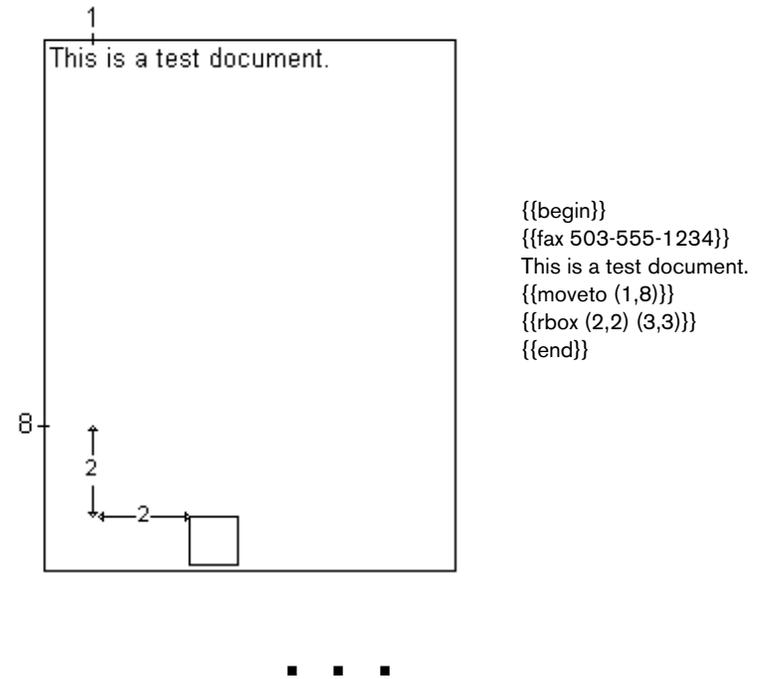
Figure 15.6 Rbox Command



The next example starts the box at (2, 2) from the current location when the Integration Module finds the `{{rbox}}` command. In this case, the location has been changed with the `{{moveto}}` command to (1, 8). Thus, the box starts (2, 2) from (1, 8), or (3, 10). The box ends at (3, 3) from (1, 8), or (4, 11).

This example shows grid numbers for the (1, 8) position. The arrows in the example document indicate that the box starts (2, 2) from the current cursor location. These arrows would not appear in the finished document.

Figure 15.7 Rbox Command



```
{{begin}}  
{{fax 503-555-1234}}  
This is a {{lineto (4, 4)}}test document.  
{{end}}
```

Chapter 16

Setting Margins and Tabs

This chapter describes the FCL commands that you can use to set margins and tabs in documents. You can set default margins for all documents, and then create single documents (or a group of documents) that override the defaults, as necessary.

Setting Margins

You can set margins for individual documents or default margins that apply to all documents. Three FCL commands control margins:

- `{{bm}}` bottom margin
- `{{lm}}` left margin
- `{{tm}}` top margin

To set margins for a single document, insert one or more of the margin commands into the FCL document.

To set default margins for all documents, insert one or more of the margin commands into a global include file. (For information on include files, see [Chapter 13, “Including the Same Files and](#)

[Commands in Many Documents](#)”.) Any document that contains a margin command will not use the defaults established in the global include file.

Example You use a global include file to establish one-inch margins for all documents. Later, you need to send a batch of 100 documents with two-inch margins. To accomplish this, you include the relevant margin commands in the FCL for those 100 documents. The 100 documents are processed with two-inch margins and all others use the default margins.

Setting margins for individual documents

To set margins for a document, insert the relevant margin commands into the FCL document.

The following example shows an FCL document with left, top, and bottom margins that are all one unit.

```
{{begin}}
{{lm 1.0}}
{{tm 1.0}}
{{bm 1.0}}
{{fax 503-555-4489}}
Body of the document to be sent.
{{end}}
```

The number you insert in the margin command reflects *current units*, so 1.0 could mean one inch, one centimeter, one point, or one pixel. This depends on the default units you have established.

You establish defaults, including default units, with the Integration Module Configuration program or by inserting certain FCL commands into a global include file. For more information on establishing default values, see [Chapter 9, “Setting Defaults for FCL Documents”](#).

For detailed information on the margin commands, including syntax and examples see [Appendix B, “FCL Commands”](#).

Setting default margins for all documents

To set default margins for all documents, insert the relevant margin commands into the global include file called Global.beg.

The following example shows default margin information. By inserting these commands into the Global.beg include file, every document that the Integration Module processes will have one-inch margins.

```
{{lm 1.0}}
{{tm 1.0}}
{{bm 1.0}}
```

Default margins are zero until you establish a different default value.

For information on using global include files, see [Chapter 13, “Including the Same Files and Commands in Many Documents”](#).

For detailed information on the margin commands, including syntax and examples, see [Appendix B, “FCL Commands”](#).

Starting a New Page With the Same Margins as the Previous Page

Use the `{{ff}}` (form feed) command to begin a new page with the same left and top margins as the previous page. Use this command in individual documents only, not in a global include file to create a default setting.

To start a new page, insert the `{{ff}}` command into the FCL.

Example The following example shows that a new page will start between the words “document” and “to.” The new page will retain the existing one-unit margins and not revert to the default margins.

```
{{begin}}
{{lm 1.0}}
{{tm 1.0}}
{{bm 1.0}}
{{fax 503-555-4489}}
Body of the document {{ff}} to be sent.
{{end}}
```

Setting Tabs with FCL

You can set tabs for individual documents or as defaults that will govern all documents. Three FCL commands control tabs:

- `{{cleartabs}}` removes all tab stops from a document.
- `{{settab}}` creates a tab stop that functions globally throughout a document.
- `{{tab}}` creates a single tab stop; can be generic (based on a preset tab stop) or can identify one of the tab stops created with the `{{settab}}` command.

To set tabs for a single document requires that you insert one or more of the three tab commands into the FCL.

To set default tabs for all documents, insert one or more instance of the `{{settab}}` command into a global include file. (For information on include files, see [Chapter 13, “Including the Same Files and Commands in Many Documents”](#).) Any document that contains a tab command will not use the defaults established in the global include file.

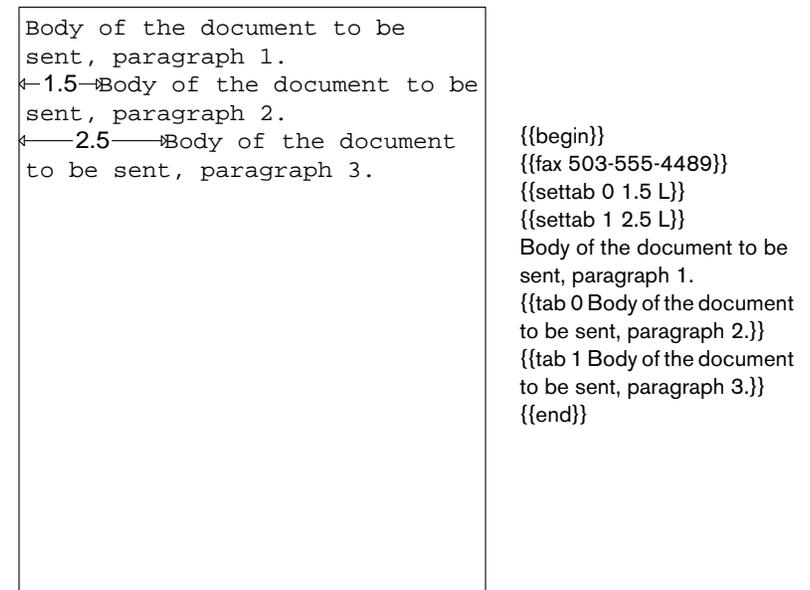
Example You use a global include file to establish tab stops at one inch, two inches, and three inches for all of the documents. Later, you need to send a batch of 100 documents with a half-inch tab stop. To accomplish this, you include the relevant tab commands in the FCL for those 100 documents. The 100 documents include a half-inch tab stop, and all others revert back to the default.

Setting tabs for individual documents

To set tabs for a document, insert the relevant commands into the FCL.

Example The following example shows an FCL document with the `{{settab}}` command creating two different tab stops. The first is labelled 0 and sets a tab stop 1.5 units from the left margin. The second is labelled 1 and sets a tab stop 2.5 units from the left margin. The two `{{tab}}` commands identify tab stops 0 and 1, and move the enclosed text accordingly.

Figure 16.1 Settab Command



Setting default tabs for all documents

1. Insert any `{{settab}}` command into the global include file called Global.beg.
2. Insert corresponding `{{tab}}` commands into the FCL for any host document.

The host document will use the tab stops described in the corresponding `{{settab}}` command.

For information on using global include files, see [Chapter 13](#), “Including the Same Files and Commands in Many Documents”.

For detailed information on the tab commands, including syntax and examples, see [Appendix B](#), “FCL Commands”.

■ ■ ■

Chapter 17

Selecting and Configuring Fonts

This chapter explains the fonts you can use with the Integration Module and explains how to manipulate fonts.

All of the installed fonts are bitmap display format (BDF) fonts. BDF fonts are not easily resized or manipulated. However, the Integration Module supports for vector fonts (such as TrueType fonts), which can be manipulated.

Fonts are proportional or non-proportional. In a proportional font family, the characters vary in width. For example, the letter I is narrower than the letter W. In a non-proportional font family, all the letters are the same width. This is apparent when text must align in vertical columns.

When it is necessary for letter alignment, use a non-proportional font. The following installed fonts are non-proportional:

- Block
- Courier
- Computer Modern Teletype
- Lucida Typewriter

The default font is Courier bold 12-point (courb12).

Understanding TrueType Fonts

The Integration Module can use any TrueType font that is supported by Microsoft Windows. TrueType font support provides flexible, customized manipulation of characters. This includes:

- Font size
- Leading (vertical space)
- Pitch (horizontal space)
- Weight
- Strikethrough
- Italics

Font properties are controlled by the `{{font}}` command. For more information on this command, including syntax and examples, see [Appendix B, “FCL Commands”](#).

Understanding Support for Other Fonts

The Integration Module also supports any fonts that are supported by Microsoft Windows.

The Integration Module software renders Windows fonts differently than the installed or TrueType fonts, and the clarity of these fonts might suffer, depending on the font. Thus, Captaris recommends that you use the installed fonts or TrueType fonts.

The Integration Module does not support double-byte fonts (such as fonts used for Chinese and Japanese).

Selecting Fonts

You can establish a font for an individual document or a default font for all documents.

To set the font for a single document:

- Use the `{{font}}` FCL command in the FCL document.
- Use the `{{font}}` command in a standard include file, as described in [Chapter 13, “Including the Same Files and Commands in Many Documents”](#).

To change the default font for all documents, use the `{{font}}` command in a global include file (as described in [Chapter 13, “Including the Same Files and Commands in Many Documents”](#).)

Documents that contains a font command will not use the defaults established in the global include file. The default font for the Integration Module is Courier bold 12-point (courb12).

Selecting fonts for individual documents

To set the font for a document, use the `{{font}}` FCL command in the FCL document.

For detailed information on the `{{font}}` command, including syntax and examples, see [Appendix B, “FCL Commands”](#).

Example The following example shows an FCL document with the font Times New Roman, 24-point, extra-bold, and italic. The `{{font}}` command controls all text until another `{{font}}` command or the end of the document.

```

{{begin}}
{{font "times new roman" size=24 extrabold italic}}
{{fax 503-555-4489}}
Body of the document to be sent.
{{end}}

```

Selecting a default font for all documents

To change the default font for all documents, use the `{{font}}` command in the global include file called Global.beg.

Example The following example shows an FCL command that will establish a default font. By inserting this command into the Global.beg include file, every document that the Integration Module processes will use the font Times New Roman, 12-point.

```

{{font "times new roman" size=12}}

```

For information on using global include files, see [Chapter 13, “Including the Same Files and Commands in Many Documents”](#).

For detailed information on the `{{font}}` command, including syntax and examples, see [Appendix B, “FCL Commands”](#).

Changing the Appearance of Fonts

You can change the appearance of fonts, such as the weight or pitch for TrueType and Windows fonts with the `{{font}}` command.

For the installed BDF fonts, you can configure only the pitch and leading.

You can underline any font with the `{{underline}}` command or draw a line through any text with the `{{font}}` with the `strikeout` option.

For detailed information on the `{{font}}` and `{{underline}}` commands, including syntax and examples, see [Appendix B, “FCL Commands”](#).

Example The following example shows the FCL commands that would change the appearance of the same font and what the finished document would look like.

Figure 17.1 Changing the Look of Fonts with FCL

<p><i>Body text, paragraph 1.</i></p> <p><u>Body text, paragraph 2.</u></p> <p>Body text, paragraph 3.</p> <p><i>Body text in Lucida Sans, 12-point, italic. This is an installed font.</i></p>	<pre>{{begin}} {{fax 503-555-8823}} {{font "times new roman" size=12 pitch=15 extrabold italic strikeout}} Body text, paragraph 1. {{underline on}} Body text, paragraph 2. {{underline off}} {{font "times new roman" size=12 regular}} Body text, paragraph 3. {{font luis12}} Body text in Lucida Sans, 12-point, italic. This is an installed font. {{end}}</pre>
---	---

■ ■ ■

Chapter 18

Setting Page Orientation and Image Quality

This chapter explains how to set page orientation (landscape or portrait) and image quality (standard or fine) using the Integration Module Configuration program or FCL.

Setting Page Orientation

To determine the page orientation for individual documents or the default for all documents, use the `{{orient}}` FCL command.

To set the default orientation for all documents, insert the `{{orient}}` command in a global include file (see [Chapter 13, “Including the Same Files and Commands in Many Documents”](#)).

Individual documents that contain the `{{orient}}` command will override the default in the global include file.

Example You use a global include file to establish landscape orientation for all documents (overriding the Integration Module default of portrait orientation). Later, you need to send a batch of 100 documents that must have portrait orientation. To accomplish this, you include the `{{orient portrait}}` command in the FCL for those 100 documents. The 100 documents are processed with portrait orientation, and all others revert back to the default.

You can change orientation multiple times in a document, so that, for example, the odd pages are portrait and the even pages landscape. You can do this with a new `{{orient}}` command at the beginning of each page (you can start a new page with the `{{ff}}` command, as described on [page 197](#)).

Setting page orientation for individual documents

To set orientation for a document, insert either `{{orient portrait}}` or `{{orient landscape}}` into the FCL document.

The following example shows an FCL document with portrait orientation.

```
{{begin}}
{{orient portrait}}
{{fax 503-555-4489}}
Body of the document to be sent.
{{end}}
```

For detailed information on the `{{orient}}` command, including syntax and examples, see [Appendix B, “FCL Commands”](#).

Setting default orientation for all documents

The default page orientation for all documents is portrait. To change this default, insert the `{{orient landscape}}` command into the global include file called `Global.beg`.

For information on using global include files, see [Chapter 13, “Including the Same Files and Commands in Many Documents”](#).

For detailed information on the `{{orient}}` command, including syntax and examples, see [Appendix B, “FCL Commands”](#).

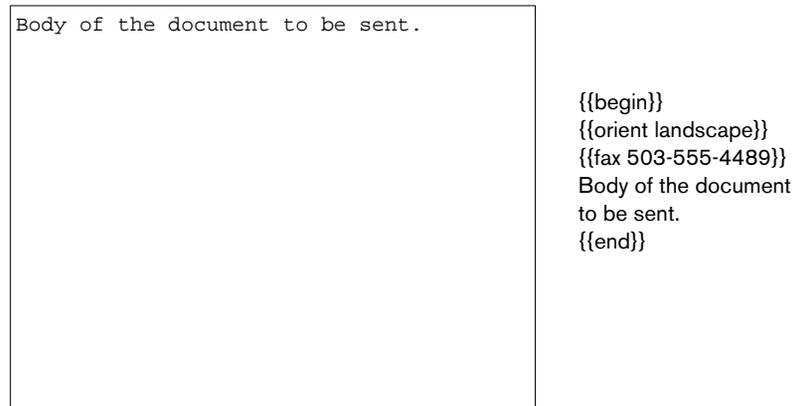
For more information on establishing default values, see [Chapter 9, “Setting Defaults for FCL Documents”](#).

Examples

This section shows two examples of how different FCL documents yield different orientation in their finished documents.

The following example shows a single document being changed from the default of portrait orientation to landscape orientation.

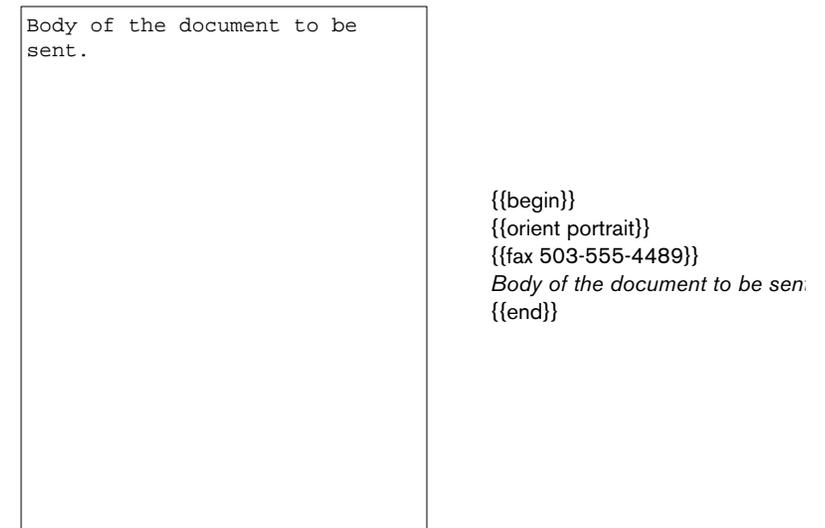
Figure 18.1 Changing to Landscape Orientation



The following example assumes that you have previously changed the default orientation to landscape by inserting the `{{orient landscape}}` command into the `Global.beg` file (a global include

file). Thus, to create a single document in portrait orientation, you must override the landscape default by inserting the `{{orient portrait}}` command in the FCL for the document.

Figure 18.2 Changing to Portrait Orientation



Setting Fax Image Quality

You can set the resolution of the fax image to fine or standard.

- Standard resolution is 204 x 98 dots per inch (dpi).
- Fine resolution is 204 x 196 dpi.

You can set image quality for individual documents or as defaults that will govern all documents.

To set image quality for a single document, insert the `{{quality}}` command into the FCL document.

To set the default image quality for all documents, see [Chapter 9, “Setting Defaults for FCL Documents”](#).

Any document that contains the `{{quality}}` command will override the default established with the Integration Module Configuration program.

Example You use the Integration Module Configuration program to establish standard image quality for all documents. Later, you need to send a batch of 100 documents that must have fine quality. To accomplish this, you include the `{{quality fine}}` command in the FCL for those 100 documents. The 100 documents are processed with fine quality, and all others use the default.

Setting image quality for individual documents

To set image quality for a document, insert either `{{quality fine}}` or `{{quality standard}}` into the FCL.

The following example shows an FCL document with fine quality.

```
{{begin}}
{{quality fine}}
{{fax 503-555-4489}}
Body of the document to be sent.
{{end}}
```

For information on the `{{quality}}` command, including syntax and examples, see [Chapter 9, “Setting Defaults for FCL Documents”](#).

■ ■ ■

Chapter 19

Attaching and Embedding Files, Signatures, and Graphics

The Integration Module provides several methods for attaching and embedding files, signatures, and graphics. To embed text in a document, see [Chapter 13, “Including the Same Files and Commands in Many Documents”](#).

You can set defaults for how the Integration Module processes attached graphics. See the options for Image Cropping in the FCL Processor settings of the Integration Module Configuration program [Chapter 9, “Setting Defaults for FCL Documents”](#).

Attaching Files to Documents with FCL

The options described here are for files that are converted to fax format (TIF images) before being attached and sent. If you want to attach files in their native format, you can purchase the InternetLink Module to send documents via e-mail. For example, fax format cannot adequately represent the content of an audio file. For instructions, see the *RightFax InternetLink Module Guide*.

To attach a file to a document, use the `{{attach}}` command. You can attach multiple documents by inserting multiple `{{attach}}` commands. The `{{attach}}` command ends the page, so the attachment always begins on its own page. The `{{attach}}` command cannot be used in `{{type email}}` documents.

To attach a library document, use the `{{libdoc}}` command. `{{Libdoc}}` has the same functionality as `{{attach}}`. For information on library documents, see the *RightFax Administrator's Guide*.

Converting attached files

Attached files are converted to fax format (.tif) in one of two ways: server-side application (SSA) file conversions and INSO conversions.

To convert a file to fax format with SSA, the source application that created the attachment must be SSA-compatible (see the *RightFax Administrator's Guide* for a list of SSA-compatible applications) and the source application must be installed on the RightFax server computer.

INSO conversions are used for files that cannot be converted with SSA. The RightFax software attempts SSA conversions for all attachments. If the attachment's file type is not SSA-compatible, or if the attachment's source application is not installed on the RightFax server computer, then an INSO conversion is used.

For more information on file conversions, see the *RightFax Administrator's Guide*.

Storing signatures

For every embedded signature file, you must either provide a path to the file in the `{{signature}}` command, or save the file in the `RightFax\Production\Forms` folder. If you provide a path, it should be a full path. If you provide a path that is not a full path, then it must be relative to `RightFax\Production\Forms`.

Embedding a signature

1. Insert the `{{signature}}` command into the FCL to define the signature file.
2. Insert `{{sign}}`, `{{signed}}`, or `{{@}}` (they are identical) at the point where you want the signature to appear.

Example In the following example, the `{{signature}}` command establishes that `DoctorRobertCribbs.tif` will be the signature used in this document. The `{{@}}` command places this signature below the text that forms the body of the document, at the current cursor location.

For this example document to function properly, `DoctorRobertCribbs.tif` must already be saved as a Class F TIF in the `RightFax\Production\Forms` folder.

```
{{begin}}{{signature DoctorRobertCribbs.tif}}
{{fax 503-555-0016}}
Body of the document to be sent.
Sincerely,
{{@}}
{{end}}
```

Because it *identifies*, rather than *places*, a signature file, the `{{signature}}` command is often useful in situations where you can manipulate the header of an FCL document, but not the body.

Example One-hundred sales people need to send a letter to 10,000 customers—each sales person must send the letter to 100 customers. The `{{signature}}` command can identify each sales person’s signature that will appear at the bottom of his or her letters.

For detailed information on the `{{signature}}`, `{{sign}}`, `{{signed}}`, or `{{@}}` commands, including syntax and examples, see [Appendix B, “FCL Commands”](#).

When you are able to manipulate the body of an FCL document, embedding a signature file might be easier when you use one of the `{{place}}` commands, which requires only one FCL command instead of two and offers more exact placement options. For more information, see [“Embedding Graphics with FCL”](#) on [page 108](#).

Embedding Graphics with FCL

To embed graphics in the body of a document, use one or more of the `{{place}}` commands.

- `{{place}}` positions the specified Class F TIF image. You can specify a full path to the graphic file. If you do not specify a path, then the default is `RightFax\Production\Forms`. The upper-left corner of the graphic is placed at the current cursor location, unless you specify `x-` and `y-` coordinates.
- `{{placeall}}` places the image on the current and all subsequent FCL pages (but not on file attachments).
- `{{placelast}}` places the image on the last page only. You can use multiple `{{placelast}}` commands to embed multiple images on the last page (such as multiple signatures, something you cannot do with the signature commands).
- `{{placexy}}` specifies the location on the page of the images defined in subsequent `{{place}}` commands. You can specify horizontal values of `left`, `center`, and `right`, and vertical values of `top`, `center`, and `bottom`. For example, `{{placexy center center}}` would align the horizontal and vertical centers of the image at the location specified in a subsequent `{{place}}` command.

Embedding a signature file ([page 106](#)) is essentially the same as embedding any other graphic. The signature commands have no functional differences from the place commands—both let you embed a graphic file in the body of a document. However, the `{{place}}` commands require less programming and let you locate the embedded graphic more precisely in the document.

■ ■ ■

Chapter 20

Scheduling Document Transmission

This chapter explains the FCL commands that schedule when the Integration Module sends documents. Unless you specify otherwise with one of these commands, the Integration Module sends documents when it receives them from the host application.

The following table lists the scheduling commands and gives a brief description.

For detailed information on these commands, including syntax and examples, see [Appendix B, “FCL Commands”](#).

Table 20a The Scheduling Commands

Command	Description
{{approval}}	Holds a document in the FaxUtil mailbox until it is approved.
{{batch}}	Holds a document so that it can be sent with other documents in a batch.
{{date}}	Specifies a future date to send a document. Combine this command with {{time}}.
{{delay}}	Delays sending by a specified number of minutes..
{{preview}}	Holds the document for preview in the FaxUtil mailbox.

Table 20a The Scheduling Commands (Continued)

Command	Description
{{priority}}	<p>Specifies a sending status of high, medium (or normal), or low priority for a document. When high priority documents exist, low and medium priority documents are held so that the high priority document can be sent. The default is low priority.</p> <p>Priority only has an effect on sending time when certain variables, such as document traffic or processing time (for complex documents with many graphics, attachments, etc.), are sufficiently high. “Sufficiently high” depends on multiple variables also, such as the number of RightFax servers or channels in the system.</p> <p>The {{batch}} command is a common variable that affects the sequence in which documents are sent, as does the priority that you establish for {{batch}} documents.</p> <p>For example, if you batch several documents and assign them all high priority, they will not be sent until the batch is complete. In this instance, low priority documents can be sent before the high priority batch documents are sent.</p>

Table 20a The Scheduling Commands (Continued)

Command	Description
{{time}}	<p>Specifies a future time to send a document. Combine this command with {{date}} to send a document at a particular time on a future day.</p> <p>The {{time}} command only considers the time remaining in the same day. Thus, a document is sent immediately if the time specified in the {{time}} command is in the past.</p> <p>For example, if you send a document to the Integration Module at 21:00 (9:00 p.m.) and you insert the command {{time 20:45}}, the document will be sent immediately because 20:45 (8:45 p.m.) of the current day is in the past.</p>
{{UTC}}	<p>Specifies a future date and time to send a document in universal coordinated time.</p>

■ ■ ■

Setting Up Notification Messages of Document Transmission

This chapter explains how to configure the Integration Module to send notification messages when a document has been sent or has errors in transmission. You have multiple options for configuring notification messages—the content they contain, how the Integration Module sends them, and how they are received.

Notification messages are informative—they can describe whether or not a document was transmitted, any transmission errors, the date and time of transmission, the owner of the document, and more. Another form of notification is to print or fax a copy of the document to another recipient such as a system administrator. For more information on this option, see [Chapter 23, “Setting Up Actions on Document Transmission”](#).

Creating Notification Messages with FCL

To send notification messages, the FCL command `{{notifyhost}}` must be included in documents sent from the host application or added to include file that is linked to the documents.

When a document is received, generated, and transmitted from the RightFax server, notification messages are generated by the program `Notify.exe` in the Integration Module. The notification messages are sent to the host application via “notification channels” that you set up in the Integration Module.

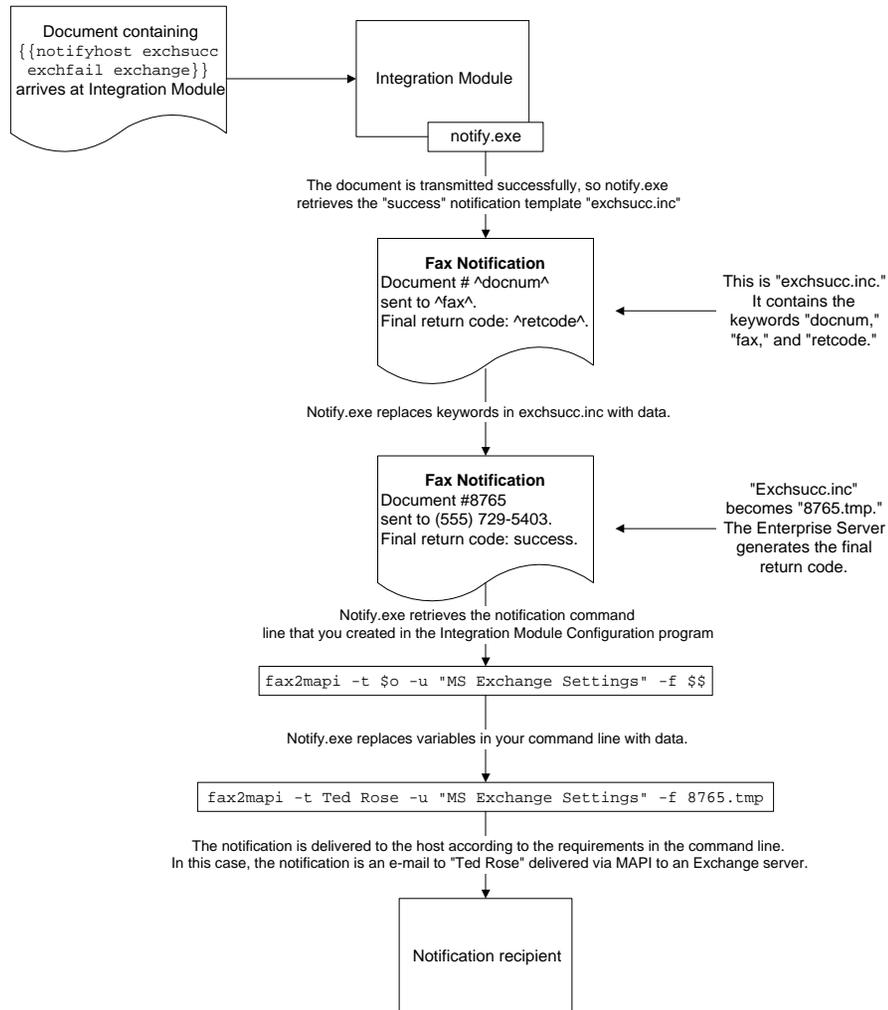
Notification messages are not generated by the Integration Module for incomplete faxes. Usually, these are faxes that are missing information that is required for sending (such as the fax number). You have the following options to set up notifications of faxes with incomplete information:

- Use Enterprise Fax Manager and FaxUtil to set up notifications and to monitor the system for faxes with incomplete information (described in the *RightFax Administrator's Guide*).
- Use the RightFax COM Module or the RightFax API to create custom notifications. Refer to the *RightFax COM Module Guide* for more information.

“[Figure 21.1: The Notification Process](#)” on [page 112](#) illustrates the FCL notification process using the following notification channel command line:

```
FAX2MAPI -t $o -u “MS Exchange Settings” -f $$
```

Figure 21.1 The Notification Process



Creating notifications for FCL documents requires these basic steps:

1. Create a template for the notification message (described on [page 112](#)). Notification templates determine the format and content of the notification message.
2. Create a notification channel (described on [page 118](#)). Use the Integration Module Configuration program to create notification channels, which include command lines that determine how the Integration Module will send notifications.
3. Include the `{{notifyhost}}` FCL command in the document that is sent from the host application to the Integration Module (described on [page 128](#)). The `{{notifyhost}}` command specifies the notification template for the message and the notification channel to use.

Creating Notification Templates

The content and format of the notification message is defined in text-based template file that you create. The template is a collection of text and keyword variables which are replaced with data about the sent document in the notification message.

Template file keywords are surrounded by carat symbols (^) and contain an identifier and an optional field length. For example, the keyword `^company^` would be replaced with the entire name of the company to which the document was sent, regardless of the number of characters. The keyword `^company 12^` would be replaced with the first 12 characters only of the company name.

For an example, refer to “[Figure 21.1: The Notification Process](#)” on [page 112](#). The template file in this example is a simple text document that contains three keywords:

- `^docnum^` is replaced with the document number assigned by the Integration Module.
- `^fax^` is replaced by the fax number of the person or company receiving the document.
- `^retcode^` is replaced with the text of the code returned from the fax board after transmission. This will either be the word “success” or a specific error code.

To create and save a notification template

1. Open Notepad or another text editing application that will produce ASCII text files.
2. Enter the text and keywords of the notification message you want to receive. For a description of all of the available keywords for notification templates, see “[Template file keywords](#)” on [page 113](#).
3. Save the file with the extension `.inc` in the `RightFax\Production\Include` folder. Make note of the file name as you will need it later when you include the `{{notifyhost}}` command in FCL documents from the host application.



Caution Do not save the template file with a `.txt` extension or any extension other than `.inc`. The Integration Module will only recognize files with this extension as notification templates.

Template file keywords

The following table lists the keywords you can add to notification template files. The keywords are shown with FCL equivalents. FCL commands in the sent document will populate the corresponding keyword in the notification message.

For example, `^altfax^` is replaced by the information included in the `{{altfax}}` FCL command in the sent document. If an FCL document contains `{{altfax 503-555-3287}}`, then 503-555-3287 will appear where the `^altfax^` keyword appears in the notification template.

Table 21a Notification Keywords and FCL Equivalents

Keyword	FCL equivalent	Description
<code>^altfax^</code>	<code>{{altfax}}</code>	The alternate fax number.
<code>^billing^</code>	<code>{{billing}}</code>	Billing code 1.
<code>^billinfo2^</code>	<code>{{billing2}}</code>	Billing code 2.
<code>^comment^</code>	<code>{{comment}}</code> <code>{{subject}}</code>	The <code>{{subject}}</code> command can be used if you have licensed the InternetLink Module. The subject appears in the subject field of the e-mail message. If no subject appears in the document, the default subject is “Success Notification” or “Error Notification.”
<code>^company^</code>	<code>{{company}}</code>	The destination company name.
<code>^contact^</code>	<code>{{contact}}</code>	The recipient name.
<code>^csi^</code>	<code>{{csi}}</code>	If no CSID (call subscriber ID) is specified, the default set in the RightFax Integration Module Configuration program will be used.
<code>^date^</code>	<code>{{date}}</code>	If no date is specified, the default is the date the fax was submitted.
<code>^dept^</code>	<code>{{dept}}</code>	The destination department number.
<code>^docnum^</code>	N/A	The document number assigned by the Integration Module.
<code>^duration^</code>	N/A	The duration of the fax call in seconds.

Table 21a Notification Keywords and FCL Equivalents (Continued)

Keyword	FCL equivalent	Description
<code>^email^</code> or <code>^mailto^</code>	{{email}} {{to}}	These commands can be used if you have licensed the RightFax InternetLink Module. The keyword <code>^email^</code> is the recipient address. With other notifications, <code>^email^</code> is typically the e-mail address of the originator of the document.
<code>^emailfrom^</code>	{{from}}	The e-mail address of the originator of the document. If no e-mail address appears in the document, the default is <code>postmaster@domain</code> . This command can be used if you have licensed the InternetLink Module.
<code>^emailcc^</code>	{{cc}}	The e-mail address where a copy of the message should be sent. This command can be used if you have licensed the InternetLink Module.
<code>^emailsubject^</code>	{{subject}}	The e-mail subject. This command can be used if you have licensed the InternetLink Module.
<code>^empid^</code>	{{empid}}	The recipient employee ID.
<code>^fax^</code>	{{fax}}	The recipient fax number. Same as <code>^phone^</code> .
<code>^phone^</code>	{{fax}}	Same as <code>^fax^</code> .
<code>^faxcard^</code>	N/A	The number of the fax board used for transmission.
<code>^faxstatus^</code>	N/A	Numeric status code; see “ The ^faxstatus^ and ^statustype^ keywords ” on page 115.
<code>^humantranstime^</code>	N/A	The date and time the fax was transmitted in human-readable format.

Table 21a Notification Keywords and FCL Equivalents (Continued)

Keyword	FCL equivalent	Description
<code>^numretry^</code>	N/A	The total number of fax attempts.
<code>^owner^</code>	{{owner}}	If no {{owner}} command appears in the sent document, the default is the RightFax user name of the owner of the fax.
<code>^ownerid^</code>	{{owner}}	The RightFax use ID of the owner of the fax
<code>^pages^</code>	N/A	Total pages of the document, not including the cover sheet.
<code>^pagessent^</code>	N/A	Total pages that were successfully transmitted.
<code>^printer^</code>	N/A	The name of the currently selected printer.
<code>^quality^</code>	{{quality}}	The fax resolution.
<code>^replyto^</code> or <code>^reply_to^</code>	{{replyto}}	The reply-to name.
<code>^retcode^</code>	N/A	The text of the code returned from the fax board after transmission. Will be either “success” or a specific error message.
<code>^rti^</code>	{{rti}}	If no RTI string appears in the sent document, the default is the RTI set in the Integration Module Configuration program.
<code>^statusstring^</code>	N/A	Fax status as listed in FaxUtil (see the <i>RightFax Administrator's Guide</i>).
<code>^statustype^</code>	N/A	Numeric return code; “ ^Statustype^ Codes ” on page 116.
<code>^termid^</code>	{{termid}}	The termination code.

Table 21a Notification Keywords and FCL Equivalents (Continued)

Keyword	FCL equivalent	Description
<code>^time^</code>	{{time}}	If no time is specified, the default is the time the fax was submitted.
<code>^tranid^</code>	{{tranid}}	The transmission ID generated by the RightFax server.
<code>^transtime^</code>	N/A	The date and time that the fax was transmitted. To specify the format for the date and time, " The ^transtime^ keyword " on page 116 .
<code>^type^</code>	{{type}}	The document transmission type (fax, e-mail, certified, etc.). This keyword is the method through which you can be notified when a document that should have been sent as a fax failed and was sent as an e-mail or certified e-mail instead. These options require the InternetLink Module (for e-mail or mime documents) or the SecureDocs module (for certified e-mail documents).
<code>^unique_id^</code>	{{unique_id}}	The unique ID for the fax assigned by the RightFax server.
<code>^user1^</code>	{{user1}}	User-defined data code 1.
<code>^user2^</code>	{{user2}}	User-defined data code 2.
<code>^user3^</code>	{{user3}}	User-defined data code 3.
<code>^userid^</code>	{{userid}}	RightFax user ID.
<code>^voice^</code>	{{voice}}	The recipient's phone number.
<code>^winsecid^</code>	{{winsecid}}	The RightFax user ID of the originator of the fax.

The `^faxstatus^` and `^statustype^` keywords

The `^faxstatus^` and `^statustype^` keywords describe the status of the sent document in the notification message. The following numeric codes will appear in the notification message.

Table 21b `^Faxstatus^` Codes

Code	Notes
0	Fax is not yet created.
1	Fax needs cover sheet.
2	Fax needs conversion.
3	Fax needs to be sent.
4	Fax is in conversion.
5	Fax needs to be sent.
6	Fax is done sending or receiving.
7	Fax uses a manual fax cover sheet.
8	Fax is scheduled to be sent.
9	Fax is done sending or receiving. Errors were encountered. Will not be retried.
10	Fax is a duplicate of another fax.
11	Error encountered. Fax will be retried.
12	Sent or received fax needs user's attention. Required data may be missing.
13	Fax needs attachment.
14	Fax is held for preview.
15	Fax is in OCR conversion.
16	Fax is printing.
17	Fax is queued for printing.
18	Fax is queued for OCR conversion.

Table 21b ^Faxstatus^ Codes

Code	Notes
19	Fax is being validated.
20	Fax is held for approval.

Table 21c ^Statustype^ Codes

Code	Status
0	Fail
2	OK

The ^transtime^ keyword

The ^transtime^ keyword and variables provide the date and time that the document was transmitted in the notification message.

You can specify the format of the date and time. The following table shows some possible formats for the transmit time 2:23 P.M. on November 6, 2001.

Table 21d Example ^Transtime^ Variables with Results

Command and variable	Result
^transtime %m/%d/%y^	11/06/01
^transtime %H:%M^	14:23
^transtime %H%M%d%Y^	14:2311062001
^transtime %B %d, %Y^	November 6, 2001

The following table lists the variables that can be used with the ^transtime^ keyword.

Table 21e ^Transtime^ Variables

Variable	Description
%a	Abbreviated weekday name.
%A	Full weekday name.
%b	Abbreviated month name.
%B	Full month name.
%c	Date and time representation appropriate for locale.
%d	Day of month as a digit (01-31).
%H	Hour in 24-hour format (00-23).
%I	Hour in 12-hour format (01-12).
%j	Day of year as a digit (001-366).
%m	Month as a digit (01-12).
%M	Minute as a digit (00-59)
%p	Current locale's A.M./P.M. indicator for 12-hour clock.
%S	Second as a digit (00-59).
%T	Local TZD.
%U	Week of year as a digit, with Sunday as first day of week (00-51). If ^transtime^ begins with %U, the time displayed is UTC (page 218) rather than local.
%w	Weekday as a digit (0-6; Sunday is 0).
%W	Week of year as decimal number, with Monday as first day of week (00-51)
%x	Date representation for current locale
%X	Time representation for current locale.

Table 21e ^Transtime^ Variables (Continued)

Variable	Description
%y	Year without century, as a digit (00-99).
%Y	Year with century, as a digit.
%z	Time-zone name or abbreviation; no characters if time zone is unknown.
%Z	Time-zone name or abbreviation; no characters if time zone is unknown.
%%	Percent.
#	Prefixes formatting code to modify as follows: %#c – long date and time %#x – long date %#d, %#H, %#l, %#j, %#m, %#M, %#S, %#U, %#w, %#W, %#y, %#Y – remove any leading zeros.

Sample notification templates

“Table 21f: Example Notification Templates” shows some typical notification template files and the resulting notification messages.

The following FCL document includes the `{{notifyhost}}` command:

```

{{begin}}
{{fax 503-555-1234}}
{{notifyhost notifysuccess.inc notifyfail.inc 1}}
{{company Acme Steel Company}}
{{contact John Smith}}
{{comment Inv. # 12345}}
{{user1 JB1234KU-6789DJJS}}
{{owner William Murray}}
Body of the document to be sent.
{{end}}
    
```

Table 21f Example Notification Templates

Template file with keywords	Resulting notification message
Duration ^duration 4^	Duration 34
Date ^transtime %m/%d/%y^	Date 05/22/2001
Comment ^comment 40^	Comment Inv. #12345
Time ^transtime %H:%M^	Time 08:37
Fax ^phone 40^	Fax 503-555-1234
Page ^pagessent 02^/^pages 02^	Page 02/02
Return Code ^retcode 20^	Return code success

Table 21f Example Notification Templates

Template file with keywords	Resulting notification message
Date ^transtime %B %d, %Y^ Sent to this number: ^fax 40^ at ^transtime %H:%M^ Regarding ^comment 40^ Sent to ^contact 40^ Sending confirmation: ^retcode 20^	Date May 22, 2001 Sent to this number: 503-555-1234 at 8:37 Regarding Inv. # 12345 Sent to John Smith Sending confirmation: success
Success/error code: ^retcode^ Sent to ^contact 40^ at ^company 60 ^ Subject: ^comment 40^ Day/time: ^transtime %B %d, %Y^/^transtime %H:%M^	Success/error code: invalid phone number Sent to John Smith at Acme Steel Company Subject: Inv. # 12345 Day/time: May 22, 2001/8:37

This command line defines how the notification will be sent. In this case, the program fax2mapi.exe will send the notification message to a MAPI-compliant e-mail box. The switches **-t**, **-u**, and **-f** are unique to Fax2mapi.exe. Each executable has its own switches.

To configure the notification command line in the Integration Module Configuration program, you use:

- Switches that the individual executable will recognize.
- Variables that will supply information about the sent document for the notification message.

To create a new notification channel

1. On the RightFax server, select **Start > Programs > RightFax > Enterprise Fax Manager**.
2. In Enterprise Fax Manager, in the **Fax Servers** list, click the name of the server on which the Integration Module is running.
3. In the **Service Name** list, double-click **RightFax Integration Module**. The **Integration Module Configuration** window appears.

Creating Notification Channels

Notifications are sent to the host application via channels. The notification channel is specified in the command line that defines how the notification will be sent to the host application. The Integration Module supports 128 notification channels.



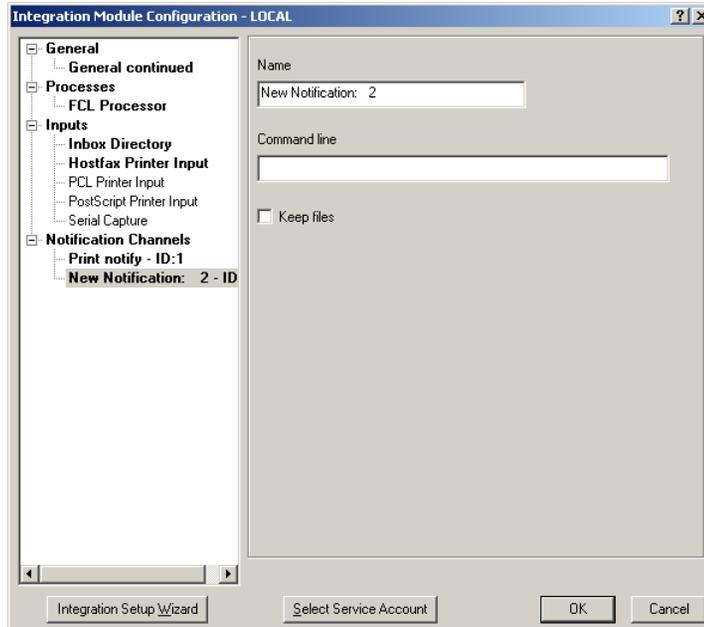
Warning Notification channel 16 is a default that the Integration Module uses when no `{{notifyhost}}` command is specified. Do not change the settings for channel 16.

“Figure 21.1: The Notification Process” on page 112 uses the following notification command line:

```
fax2mapi -t $o -u "MS Exchange Settings" -f $$
```

- Right-click **Notification Channels**, and select **Add Output Device** from the shortcut menu. The available settings for the notification channel appear in the **Integration Module Configuration** window.

Figure 21.2 The Notification Channels settings



- In the **Name** box, enter a descriptive name for the channel. The channel number is assigned as you add channels.

- In the **Command line** box, enter the case-sensitive command line for the channel. For information on the command line to use for each type of notification, refer to the appropriate section in this chapter:

- [“Sending notifications via SMTP” on page 121](#)
- [“Sending notifications to a database” on page 122](#)
- [“Sending notifications to an SMS-enabled device” on page 123](#)
- [“Sending notifications via 3270 emulation” on page 124](#)
- [“Sending notifications via FTP” on page 125](#)
- [“Sending notifications via IBM WebSphere MQ” on page 126](#)
- [“Sending notifications to Lotus Notes” on page 127](#)
- [“Sending notifications to Microsoft Exchange” on page 127](#)

- Select the **Keep files** check box if you want to save a copy of each notification that is sent to the host application. Notifications are saved in the Windows Temp folder.

- Click **OK** to save the new notification channel.

Methods used for notification messages

The following table lists the common methods you can use for connecting to and sending notification messages to a device or host application. Contact Captaris for information on other available notification methods.

Notify.exe on the RightFax Integration Module sends the notification message to the host application via the method listed in the first column using the executable listed in the second column.

Table 21g Notification Methods and Executable Program Files

Connection method	Program that sends the notification	Description
SMTP	Mailsend.exe	Sends messages via SMTP to an SMTP-compliant mail server.
ODBC database	Dbnotify.exe	Sends notifications to an ODBC data source.
SMS	Rfsms.exe	Sends notifications to an SMS-enabled device via the Captaris Push-Proxy Gateway (PPG) server.
3270 emulation	HlpiSEND.exe	Sends notifications back through 3270; uses high-level-language application-programming interface (HLLAPI). This requires third-party 3270 emulation software.
FTP	Ftpit.bat	Is a Perl script used to log on to an FTP server and transfer files.
IBM WebSphere MQ	Mqput.exe	Sends a message to the specified remote queue by the specified remote queue manager. This requires IBM WebSphere MQ software.
Notes	Fax2note.exe	Sends messages to a Lotus Notes system using Notes API.
Exchange	Fax2mapi.exe	Sends messages to Microsoft Exchange using MAPI. This requires a Microsoft Exchange client on the fax server.

Each of these executable files including their command line parameters are defined in the sections that follow. For each executable, one or more command line parameters support variables that let you pull information directly from the FCL in the document that was sent.

The following table lists the command line variables that can be used with all notification executables. The FCL equivalents are FCL-based information that populates the corresponding variable. For example, \$^ is replaced by the information included in the {{termid}} FCL command in a sent document. So, if an FCL document contains {{termid A3}}, then A3 will be used in the command when it executes.

Table 21h Notification Channel Variables

Variable	FCL equivalent	Notes
\$\$	N/A	File name of the notification created by Notify.exe. This file contains the text of the notification to be sent and is formatted using the notification template specified with the {{notifyhost}} FCL command.
\$^	{{termid}}	The termination code of the sent fax.
\$B	{{billing}} {{billing2}}	Billing codes 1 and 2
\$C	{{comment}}	Comment text.
\$c	{{emailcc}}	The e-mail address where a copy of the message should be sent. This command can be used if you have licensed the InternetLink Module.
\$d	N/A	The document number assigned by the Integration Module.
\$D	{{dept}}	The recipient's department.
\$E	{{empid}}	The employee ID.

Table 21h Notification Channel Variables (Continued)

Variable	FCL equivalent	Notes
\$f	{{from}}	The e-mail address of the originator of the document. This command can be used if you have licensed the InternetLink Module.
\$o	{{owner}}	The RightFax ID of the originator of the fax.
\$p	N/A	The name of the currently selected printer.
\$P	N/A	The number of pages sent, not including the cover sheet.
\$r	N/A	The return code.
\$R	{{replyto}}	The reply to name.
\$s	{{subject}}	The e-mail subject. This command can be used if you have licensed the InternetLink Module.
\$t	{{to}}	The e-mail address of the recipient of the message. This command can be used if you have licensed the InternetLink Module.
\$T	N/A	The TIF file name of the document.
\$w	{{winsecid}}	The RightFax user ID of the originator of the fax.
\$1	{{user1}}	User-defined data code 1.
\$2	{{user2}}	User-defined data code 2.
\$3	{{user3}}	User-defined data code 3.

Sending notifications via SMTP

To create a notification channel that sends messages to an SMTP mail server, use Mailsend.exe.

Syntax `mailsend [options] input filename`

Table 21i Mailsend.exe Command Line Options

Option	Description
-a	Abort if connect fails.
-c <i>address</i>	E-mail address where a copy of the message will be sent. The e-mail address can appear in the following formats: Ashutosh Apte <AshutoshApte@captaris.com> <AshutoshApte@captaris.com> AshutoshApte@captaris.com Mailsend.exe will convert these formats to <AshutoshApte@captaris.com>.
-f <i>address</i>	E-mail address of the sender of the document.
-h	Displays online help for Mailsend.exe.
-H	Input file contains mail headers.
-m " <i>mailhost</i> "	The name of the mail host where messages will be sent. The name must be surrounded by quotation marks.
-o	Obtain the recipient e-mail address from the first line of the file, and obtain the subject of the e-mail message from the second line.
-s <i>subject</i>	Subject of the e-mail message.

Table 21i Mailsend.exe Command Line Options (Continued)

Option	Description
-t <i>address</i>	The e-mail address of the recipient of the notification message. The e-mail address can appear in the following formats: Ashutosh Apte <AshutoshApte@captaris.com> <AshutoshApte@captaris.com> AshutoshApte@captaris.com Mailsend.exe will convert these formats to <AshutoshApte@captaris.com>.
-v	Display verbose messages.

Example mailsend -m "smtpserver.yourhost.com" -t "\$o" -f RightFaxAdmin@company.com -s "Notification of Fax" \$\$

Table 21j Description of the Example Command Line

Element3	Description
mailsend	The name of the executable file that will process the notification.
-m "smtpserver.yourhost.com"	The name of the mail host.
-t "\$o"	The recipient's name, who is also the sender of the document. The recipient will be obtained from the {{owner}} FCL command in the document received from the host application.
-f RightFaxAdmin@company.com	The e-mail system's address and the sender's SMTP address, which is required in order to receive the notification.

Table 21j Description of the Example Command Line

Element3	Description
-s "Notification of Fax"	The text that will appear in the subject line of the e-mail message.
\$\$	The file name of the document.

Sending notifications to a database

To create a notification channel that sends SQL messages to a database, follow these steps:

1. Set up a new Data Source Name (DSN) by selecting **Start > Settings > Control Panel > Data Sources (ODBC)**. This opens the **ODBC Data Source Administrator** dialog box. Click **Add** and complete the options to create the new DSN. Make a note of the DSN name as you will need it later.
2. Create notification templates for both success and failure notifications. These are text files that contain the SQL commands that define where and how the notifications will be written to the database.

Example Insert SampleTable(ID,Status)
Values(^DocNum^, '^RetCode^')

- Configure a new notification channel (described on [page 118](#)) using the Dbnotify.exe program in the **Command Line** box.

Syntax `dbnotify {-fFileName|-sSQL} [-uUserID] [-pPassword] DSN`

Table 21k Dbnotify.exe Command Line Options

Option	Description
-fFileName	Fully qualified file name of a file containing the SQL script for the notification. You must use -f or -s, but not both.
-sSQL	SQL script to execute containing the notification script. You must use -f or -s, but not both.
-uUserID	User ID. This is only required if your ODBC data source was not configured with a user name.
-pPassword	Password. This is only required if your ODBC data source was not configured with a password.
DSN	Data Source Name that you created in step 1.

Example `dbnotify -f$$ DSN`

Table 21l Description of the Example Command Line Above

Element3	Description
dbnotify	The name of the executable file that will process the notification.
-f \$\$	Tells Dbnotify to use the SQL script in the template that was specified in the {{notifyhost}} command in the data stream.
DSN	This is the name of the DSN you created in step 1.

Sending notifications to an SMS-enabled device

To send notifications to an SMS-enabled device via the Captaris Push-Proxy Gateway (PPG) server, follow these steps:

- Install and configure the Captaris Push-Proxy Gateway server. Refer to the documentation for this product for instructions.
- Create a Pager/SMS service in Enterprise Fax Manager with SMS as the **Service Type**. This service is required for communication between the RightFax server and the Push-Proxy Gateway server that sends the SMS messages. For information on creating SMS/Pager services in Enterprise Fax Manager, refer to the *RightFax Administrator's Guide*.
- Create notification templates for both success and failure notifications (described on [page 112](#)). The total length of the notification message, including the text substituted for keyword variables may not exceed 160 characters (the maximum allowed length for SMS messages).
- Configure a new notification channel (described on [page 118](#)) using the Rfsms.exe program in the **Command Line** box.

Syntax `rfsms -sSMSService -dSMSNumber [{-mFile}"MessageText"] [-v] [-fRFServer] [-uUserID] [-pPassword]`



Caution *Rfsms.exe does not perform error checking on the parameters you enter on the command line. Make sure that all command line parameters are correct and that the accounts specified exist on the server.*

Table 21m Rfsms.exe Command Line Options

Option	Description
-sSMSService	The Service ID of the SMS/Pager service you created in Enterprise Fax Manager.
-dSMSNumber	The phone number of the SMS device you will be contacting.

Table 21m Rfsm.exe Command Line Options (Continued)

Option	Description
{-mFile} "MessageText"}	Specify a file name containing the alert text, or enter the alert text between quotes. If you specify the message text in quotes, do not use the -m switch.
-v	Enables verbose event logging.
[-fRFServer]	The name of the RightFax server on which the SMS service specified with the -s option resides. This is only required if the SMS service is not on the current server.
[-uUserID]	The RightFax user ID required to log on to the RightFax server. This is only required if you are not using a trusted account.
[-pPassword]	The password for the RightFax user ID specified with the -u option.

Example rfsm.exe -fRFServer -uAdministrator -pPassword -sRFPPG -d\$1 -m\$\$

Table 21n Description of the Example Command Line Above

Element3	Description
rfsm.exe	The name of the executable file that will process the notification.
-fRFServer	The name of the RightFax server on which the specified Pager/SMS service resides.
-uAdministrator	The RightFax user ID used to log on to the server.
-pPassword	The password for the specified user ID.
-sRFPPG	The service ID of the Pager/SMS service in RightFax that sends SMS messages.

Table 21n Description of the Example Command Line Above (Continued)

Element3	Description
-d\$1	Maps to the {{SMS}} FCL code.
-m\$\$	The file containing the alert text to send.

Sending notifications via 3270 emulation

To create a notification channel that sends messages via 3270 using high level language application programming interface (HLLAPI), use Hllpise.exe.

Syntax hllpise [options] filename

Table 21o Hllpise.exe Command Line Options

Option	Description
-c	Do not clear screen before notifying.
-e character	Set escape character.
-E	Do not send enter after sending file.
-h	Displays online help for Hllpise.exe.
-H number	HLLAPI emulation package 1=Attachmate (default) 2=WRQ 3=Rumba 4=IBM Personal Com 5=NetSoft Elite (not supported)
-i	Interact with host (filename is a script).
-l script	Specify login script.
-p delay	Set host power-up delay.
-S	SSCP invalid for input.
-s number	Set session number.
-t	Use file transfer.

Table 21o Hlpisend.exe Command Line Options (Continued)

Option	Description
-v	Display verbose messages.
-w <i>seconds</i>	Specify wait after input.

Example hlpisend -s A -H 2 -l login.inc \$\$

Table 21p Description of the Example Command Line

Element	Description
hlpisend	The name of the executable file that will process the notification.
-s A	The HLLAPI shortname of the session. A is the A session. B is the B session, etc.
-H 2	The number for the HLLAPI emulation package. In this case, 2 = WRQ.
-l login.inc	The file name of the login script for the session. Login.inc is created custom for every install.
\$\$	The file name of the document.

Sending notifications via FTP

To create a notification channel that sends messages via FTP, use Ftpit.bat.

This program uses a Perl script used to log on to an FTP server and transfer files. In the command line, you must supply a host name, user name, and password. Ftpit.bat creates a remote file name called jcl###, or you can specify the file name in the command line.

Syntax ftpit [*options*] *hostname username password localfile*

Table 21q Ftpit.bat Command Line Options

Option	Description
-h	Display online help for Ftpit.bat.
-A <i>account</i>	Account name.
-a <i>file name</i>	Remote file to append to.
-c <i>command</i>	Quote command (such as LRECL (80)).
-r <i>directory</i>	Remote destination directory.

Example ftpit yourhost.com franklins qwerty \$\$

Table 21r Description of the Example Command Line

Element	Description
ftpit	The name of the executable file that will process the notification.
yourhost.com	The name of the FTP host to which the notification will be sent.
franklins	The name of the user to whom the notification will be sent.
qwerty	The password for the user and host.
\$\$	The file name of the document.

Sending notifications via IBM WebSphere MQ

To create a notification channel that sends messages via IBM WebSphere MQ, use Mqput.exe.

This program submits a message to a specified remote queue. Mqput.exe receives the body of the message from standard input (STDIN) or from a file that you specify in the command line.

Syntax `mqput -C channel -H hostname -M queue manager -Q queue [options]`

Table 21s Mqput.exe Command Line Options

Option	Description
-C <i>channel</i>	Name of the IBM WebSphere MQ channel to use for connection.
-H <i>hostname</i>	Fully qualified domain name of the IBM WebSphere MQ queue manager.
-M <i>queue manager</i>	IBM WebSphere MQ queue manager to connect to.
-Q <i>queue</i>	IBM WebSphere MQ queue to retrieve messages from.
-d	Display debugging output. This is helpful if you experience difficulty connecting to the server.
-e <i>days</i>	Days until message expires. Default is no expiration.
-f	If the message fails in delivery to the destination queue, cancel it rather than put it in the dead-letter queue. This is helpful during testing or initial setup.
-i <i>input</i>	Source of data to send to the queue. Input can be a file or standard input (STDIN). If no file is specified, STDIN is the default.
-p <i>port</i>	TCP/IP port number to use for remote connection. The default is 1414.
-r <i>priority</i>	Message priority. The default is 0.

Table 21s Mqput.exe Command Line Options (Continued)

Option	Description
-s	Make messages persistent. If the server is restarted, the IBM WebSphere MQ server will store messages so that they can be accessed after the server is restarted.
-v	Display version information.
-1	Selects Version 1 of the WebSphere MQ Application Programming Reference. This option must be used because MQPut.exe is not designed to work with the WebSphere MQ API Version 2.
-tCCSID	Specifies the codeset name for a language. A list of the codeset IDs (CCSIDs) supported by WebSphere MQ is available from IBM.

Example `mqput -C RF_Chan -H qmmaster2 -M RightFax -Q RF_Notify -i $$ -p 1414 -1`

Table 21t Description of the Example Command Line

Element	Description
mqput	The name of the executable file that will process the notification.
-C RF_Chan	The IBM WebSphere MQ channel name.
-H qmmaster2	Indicates that the fully qualified domain name of the IBM WebSphere MQ queue manager will come next.
-M RightFax	The IBM WebSphere MQ queue manager.
-Q RF_Notify	The IBM WebSphere MQ queue name.
-i \$\$	The file name of the document.
-p 1414	The TCP/IP port number.

Sending notifications to Lotus Notes

To create a notification channel that sends messages to a Lotus Notes system using the Notes API, use Fax2note.exe.

Syntax fax2note [*options*] *recipient*

Table 21u Fax2note.exe Command Line Options

Option	Description
-t <i>e-mail address</i>	Recipient(s). Multiple recipients must be separated by a semicolon (;) and enclosed in quotation marks ("). Recipients must not be ambiguous in default address book.
-s	The subject line of the e-mail message.
-a	Identifies a file to attach to the e-mail message.
-f	Identifies an ASCII text to use as the body of the e-mail message.
-o	Identifies the owner of the e-mail message.

Example fax2note -t "\$o" -s "Fax Notification" -f \$\$

Table 21v Description of the Example Command Line

Element	Description
fax2note	The name of the executable file that will process the notification.
-t "\$o"	The recipient of the notification. The recipient will be obtained from the {{owner}} FCL command in the document received from the host application.
-s "Fax Notification"	The subject line of the notification message.
-f \$\$	The file to be used as the message body. The content of the notification template file will replace the \$\$ variable.

Sending notifications to Microsoft Exchange

To create a notification channel that sends messages to an Exchange system using MAPI, use Fax2mapi.exe.

Syntax fax2mapi [*options*]

Table 21w Fax2mapi.exe Command Line Options

Option	Description
-a <i>file name</i>	Attach the specified file.
-f <i>file name</i>	Name of an ASCII text file to use as the message body.
-i <i>file name</i>	Name of an ASCII text file to use as the message body.
-k	Keep sent mail.
-p <i>password</i>	Profile password.
-q <i>CSID</i>	CSID associated with the FAX address type of recipient. This option overrides any address specified with -t.
-r <i>DID</i>	DID associated with the FAX address type of recipient. This option overrides any address specified with -t.
-s <i>subject</i>	Subject line of the e-mail message.
-t <i>e-mail address</i>	E-mail address for recipient(s). Multiple recipients must be separated by a semicolon (;) and enclosed in quotation marks ("). Recipients must not be ambiguous in default address book. The -q and -r options override the -t option.
-u <i>profile</i>	Mail profile name.
-z	File (in RTF format) to be used as a message body.

Example fax2mapi -t "\$o" -s "Fax Notification" -f \$\$ -u "MS Exchange Settings"

Table 21x Description of the Example Command Line

Element	Description
fax2mapi	The name of the executable file that will process the notification.
-t "\$0"	The recipient of the notification. The recipient will be obtained from the {{owner}} FCL command in the document received from the host application.
-s "Fax Notification"	The subject line of the notification message.
-f \$\$	The file to be used as the message body. The content of the notification template file will replace the \$\$ variable.
-u "MS Exchange Settings"	The mail profile name.

Including the {{Notifyhost}} Command in Documents

The third step in creating notifications is to include the {{notifyhost}} FCL command in the document from the host application. This command specifies the notification template. It also specifies the notification channel, as created in the RightFax Integration Module Configuration program.

In the syntax shown below, the template files (Success.inc for successful documents and Failure.inc for failed documents) are specified. If the template name is not specified, then no notification

is sent. The channel keyword specifies the channel. The channel can either be an channel number (from 1 to 128) or name. If no channel is specified, the default channel will be used, number 16.

Syntax `{{notifyhost success.inc failure.inc channel}}`

Examples `{{notifyhost mysucc myfail mynotify}}`
`{{notifyhost none myfail mynotify}}`

The first example shows that if the document is sent successfully, the Integration Module will create a notification based on a template called "mysucc." The ".inc" extension is the default, as is the path to the Include folder. If the document fails to send properly, the "myfail" template is used.

In either case, the notification is sent via the channel that is named "mynotify."

The second example sends a notification (based on the template Myfail.inc) if the document fails to send. Because "none" is entered instead of the success template name, no message will be sent if the document is transmitted successfully.

The {{notifyhost}} command can be included in the document from the host application to the Integration Module, or you can insert it in an include file. For information on include files, see [Chapter 13](#), "Including the Same Files and Commands in Many Documents".

Creating the {{Notifyhost}} command

The {{notifyhost}} command must contain:

- The name of the success template, as described in "[Creating Notification Templates](#)" on [page 112](#).
- The name of the failure template, as described in "[Creating Notification Templates](#)" on [page 112](#).
- The notification channel number or name, as described in "[Creating Notification Channels](#)" on [page 118](#).

The following example shows an FCL document that includes a {{notifyhost}} command.

```

{{begin}}
{{fax 503-555-1234}}
{{notifyhost notifysuccess.inc notifyfail.inc 1}}
{{company Acme Steel Company}}
{{contact John Smith}}
{{comment Inv. # 12345}}
{{user1 JB1234KU-6789DJJS}}
{{owner William Murray}}
{{end}}

```

Other FCL used in notifications

A number of informational FCL commands can be used for notifications. These commands store information about the sender of the document so that notification messages can be sent specifically to that person, department, terminal, or another destination.

The following table lists these commands and gives a brief explanation. For more detailed information on each of the commands, see [Appendix B, “FCL Commands”](#). For a comparison of the commands listed here and the keywords that you can use in notification templates, see [“Table 21a: Notification Keywords and FCL Equivalents”](#).

Table 21y Notification FCL Commands

Command	Description
{{Billing}}	The billing code of the document owner. This command is sometimes used to populate variables in cover sheets.
{{Billing2}}	A secondary billing code of the document owner. This command is sometimes used to populate variables in cover sheets.

Table 21y Notification FCL Commands (Continued)

Command	Description
{{Comment}}	Any user-defined message specific to the document. This command is most often used to populate variables in cover sheets but is sometimes used with notifications.
{{Company}}	The company name for the document. This command is most often used to populate variables in cover sheets but is sometimes used with notifications.
{{Contact}}	The contact name for the document. This command is most often used to populate variables in cover sheets but is sometimes used with notifications.
{{CSI}}	The CSID. This is usually the general fax number for the enterprise. You can set a default CSID in the Integration Module Configuration program. See Chapter 9, “Setting Defaults for FCL Documents” .
{{Dept}}	The department of the fax owner.
{{Email}}	The e-mail address for the recipient of the notification message.
{{EmpID}}	The employee ID of the fax owner. This command is sometimes used to populate variables in cover sheets.
{{Owner}}	The document owner's name. This command is most often used to populate variables in cover sheets but is sometimes used with notifications.

Table 21y Notification FCL Commands (Continued)

Command	Description
{{ReplyTo}}	The recipient for the notification. You can request that an HTTP post be sent back to the host as a notification when you use the RightFax XML Interface. ReplyTo is the field in the submit post that the RightFax XML Interface populates to determine where to send the notification.
{{TermID}}	The ID of the terminal from which the document originated. This command is sometimes used to populate variables in cover sheets.
{{TranID}}	The ID of the transaction that produced the document. This command is sometimes used to populate variables in cover sheets.
{{UniqueID}}	An identification number for each destination (fax number) within the document. This command is used most often for tracking. The Integration Module will generate a UniqueID unless you specify one in the FCL. Then, you can track the document in FaxUtil based on the UniqueID. Secondarily, this command is sometimes used in cover sheets and with notifications.
{{User1}} {{User2}} {{User3}}	User-defined data, such as the originator of the document (person, group, or other information).

Table 21y Notification FCL Commands (Continued)

Command	Description
{{UserID}}	The RightFax user ID of the creator of this document. This command is sometimes used to populate variables in cover sheets.
{{Voice}}	A voice telephone number. This command is sometimes used to populate variables in cover sheets.

■ ■ ■

Testing and Troubleshooting Notification Messages

This chapter describes methods for testing that the Integration Module is correctly sending notifications to the host application using the notification channels that are described in [Chapter 23](#), “[Setting Up Actions on Document Transmission](#)”.

The simplest way to test that notifications are being received properly is to send a test document. If the correct notification is received using the correct channel, then the channel is configured correctly. If not, then you can use this chapter to troubleshoot the problem.

Note that notification messages are not generated by the Integration Module for incomplete faxes. Usually, such faxes are missing information that is required for sending. For more information, see “[Creating Notification Messages with FCL](#)” on [page 111](#).

Testing That the Host Application Is Correctly Receiving Notifications

The following procedure requires that you have already created a notification channel, created a notification template, and created the `notifyhost` command that will identify the channel and template. For information on these procedures, see [Chapter 21](#), “[Setting Up Notification Messages of Document Transmission](#)”.

The information presented here is not specific to the various notification methods you might have already created. Rather, this procedure is presented in general terms that can be applied to any notification type. If you are unable to successfully troubleshoot failed notifications, you can contact Captaris Customer Support.

1. Send a test document from the host to the Integration Module.

You might use the `fax` command to have the Integration Module send the document. Besides `fax`, the test document must at least include `begin`, `end`, the correct `notifyhost` command, and some sample text.

2. Verify that you receive a notification from the Integration Module in the location that you specified when you created the notification channel.

3. Verify that you receive a notification that contains the correct information (it should be the same as the notification template that you created).

If you do not receive a notification, or if the notification does not contain the correct information, then see “[Troubleshooting](#)” on [page 132](#).

Testing IBM WebSphere MQ connections

This procedure applies to IBM WebSphere MQ connections. It verifies that data is flowing from the Integration Module to the host (as opposed to verifying that the host received a notification).

This procedure requires that you have correctly configured:

- The host to receive data from the Integration Module
- The Mqput.exe notification channel on the Integration Module (see “[Sending notifications via IBM WebSphere MQ](#)” on page 126)

To test the Integration Module-to-Host connection for IBM WebSphere MQ

1. Open a command prompt window.
2. Enter the command you entered in the **Command line** box in the **Add Output Device** dialog box when you created the notification channel.



Warning Command line options are case-sensitive. If you do not enter the command here exactly as you did in the **Add Output Device dialog box**, errors will occur.

3. Press ENTER.

Figure 22.1 Testing IBM WebSphere MQ Notifications

```

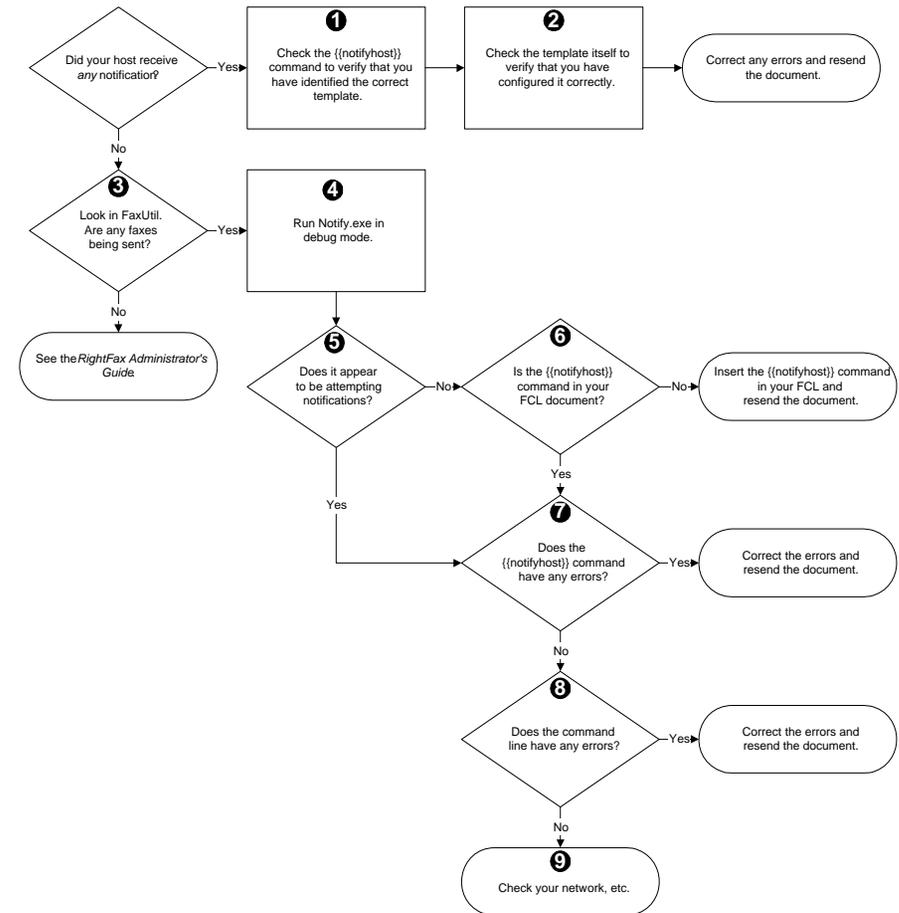
C:\WINNT\System32\cmd.exe
C:\>MQPUT.EXE -C RF_Chan -H QNMASTER2 -M RightFAX -Q RF_Queue -i C:\TEST.FCL -d
03/28 13:38:40 Using Input Source 'C:\TEST.FCL'.
03/28 13:38:40 No TCP/IP Port Specified. Using Default of '1414'.
03/28 13:38:40 Connected to 'RightFAX' <0, 0>.
03/28 13:38:40 Opened Queue 'RF_Queue' <0, 0>.
03/28 13:38:40 Put Message of length 49 <0, 0>.
03/28 13:38:40 Closed the Queue 'RF_Queue' <0, 0>.
03/28 13:38:40 Closed the Queue Manager 'RightFAX' <0, 0>.
C:\>_
    
```

If the notification channel connection is successful, then you should see messages similar to those shown in “[Figure 22.1: Testing IBM WebSphere MQ Notifications](#)”. If the output connection is not successful, then you will see error messages.

Troubleshooting

Use the following flowchart and the corresponding numbered paragraphs on the next page to identify the source of the error.

Figure 22.2 Troubleshooting Notifications



1. If you have created several templates with similar names, a minor typographical error can identify the wrong template, resulting in a notification that contains errors and nonpopulated keywords. If you specify a template that does not exist, then no notification can be created or sent.
2. The template must contain valid (and correctly spelled) keywords. For a list of valid keywords, “[Table 21 a: Notification Keywords and FCL Equivalents](#)” on [page 113](#)
3. When you do not receive a notification after sending a document, the first step should be to verify that the RightFax server is sending data. Open FaxUtil and verify that faxes are being sent. If no faxes are being sent, then no notifications can be sent.
4. If you verified in FaxUtil that faxes are being sent, then run Notify.exe in debug mode to proceed.
 - Stop Notify.exe from the **Process** tab in Windows Task Manager.
 - Open a command prompt window.
 - Type **notify.exe**, and then press ENTER.

The information that appears will indicate the status of Notify.exe.

5. When Notify.exe is attempting to send a notification, you will see information in the command prompt window that describes the error (see the next steps for more information). If Notify.exe returns a message such as “Checking no record,” then it is not receiving a signal that a document has arrived at the Integration Module. In this case, you must check the connection (see [Chapter 7, “Testing a Connection and an Input Device”](#)).

6. One reason that Notify.exe would not appear to be attempting notifications is that the {{notifyhost}} command is absent from the host data stream. The error in this case is commonly that the command exists but is misspelled, is missing one or more of its braces, or contains another syntax error that causes Notify.exe to not identify it. When this is the case, Notify.exe will return a message such as “Check Notify <ID number>” with no more text after it. This message means that Notify.exe is checking its queues; when no more text comes after this message, then Notify.exe found no queued notifications.
7. If Notify.exe appears to be attempting a notification, examine the {{notifyhost}} command in the FCL document. Common problems include:
 - A template that does not exist (Notify.exe will return a message such as “Unable to include file: <path>.inc”).
 - A notification ID that does not exist (Notify.exe will return a message such as “Cannot find notify channel [#], defaulting to [#]” or “Executing Copy C:\temp\<document number>.NT1 NUL”). In the case of the second message, the file is being copied to NUL (being deleted). This happens because Notify.exe uses the default ID (16) when no ID exists. The default action for ID 16 is to copy to NUL; thus, if you leave ID 16 unchanged, then files that are being copied to NUL probably have no ID associated with them. If you have changed the action for ID 16 (Captaris recommends that you do not do this), then nonexistent IDs will result in the action that you specified for ID 16.

8. Check the command line you created for the notification channel (see [“Creating Notification Channels”](#) on page 118). Common problems include:
 - A misspelled or invalid executable file, or (if you specified a path to the executable) the wrong path. (You will see a message such as “The name specified is not recognized as an internal or external command, operable program, or batch file”).
 - Omitted or wrong variables or switches; incorrect syntax (Notify.exe will return a message specific to the executable or command that you used).
 - In some cases, depending on the command, executable, or other command line elements, Notify.exe will not return an error message even though the command line contains an error.
9. If you have not yet found an explanation for the problem, it is likely that the Integration Module is processing and sending the notification correctly. The error is probably somewhere else, such as the network, host computer, or host application. Further evidence would be Notify.exe returning a message such as “1 file(s) copied. Function returned 0: success” (or a similar message indicating a successful action based on the executable or command you used in the command line). This means that Notify.exe has sent the notification. If you did not receive it, then the error is probably outside of the Integration Module.

■ ■ ■

Setting Up Actions on Document Transmission

This chapter explains how to configure the Integration Module to notify you when a document has been sent or has errors in transmission by setting an action, such as printing or faxing a copy of the document to another recipient. This can serve as a notification that the document needs attention.

These actions are similar to notification messages (described in [Chapter 21, “Setting Up Notification Messages of Document Transmission”](#)). Notification messages are informative—they can describe whether or not a document was transmitted, transmission errors, and other information such as the date and time of transmission and the owner of the document.

Printing or Faxing a Copy of a Document That Was Transmitted

You have two options for establishing actions that occur when documents are transmitted successfully:

- Set defaults in the Integration Module Configuration program.
- Use the `{{onsuccess}}` FCL command in a document to override the default.

Setting the default “success” action

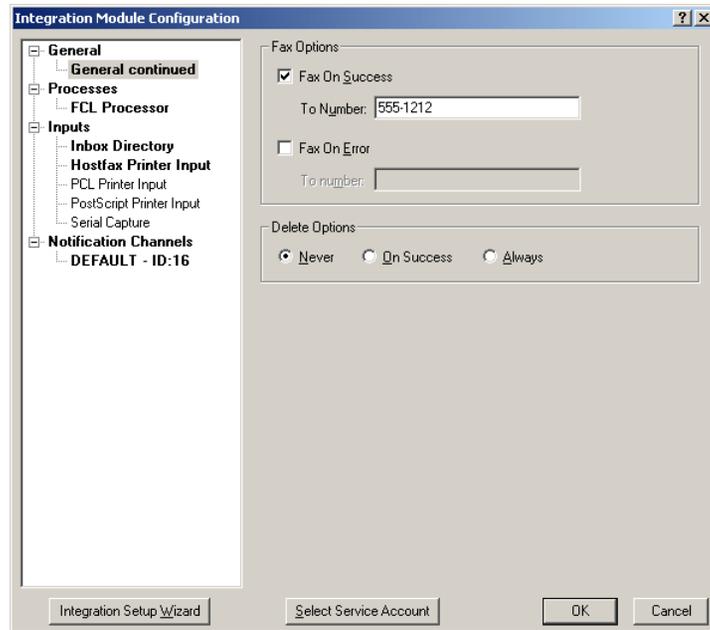
When a document is successfully transmitted, a copy of the document can be sent to another recipient as a notification of the transmission.

To set the success action

1. From the **Start** menu, select **Programs > RightFax > Enterprise Fax Manager**. The **Enterprise Fax Manager** window appears.
2. In the **Fax Servers** list, click the name of the server on which the Integration Module is running.
3. In the **Service Name** list, double-click **RightFax Integration Module**. The **Integration Module Configuration** window appears.

- In the left pane, click **General continued**.

Figure 23.1 The General Continued Settings



- Under **Fax Options**, select **Fax on Success** or **Fax on Error** to send the fax to a new recipient on these events. If you select either of these options, you must also enter a destination fax number in the **To Number** field.



Note The document also can be e-mailed or sent for certified delivery using the FCL commands `{{type email}}`, `{{type certified}}`, or `{{type mime}}`. For more information, refer to [Chapter 11, “Specifying the Destination of the Document”](#).

- Under **Delete Options**, select the option corresponding to when you want the fax images deleted from the server.

Never	Fax images will never be automatically deleted from the server.
On Success	Fax images will be automatically deleted from the server only when they have been sent successfully.
Always	Fax images will be automatically deleted from the server on both successful and failed transmission attempts.

- Click **OK**.

Using the `{{onsuccess}}` command to set “success” actions for a document

To set “success” actions for a document and override the default, add the `{{onsuccess}}` command to the FCL. In the following example, a copy of the document will be faxed to another recipient if it is successfully transmitted.

```

{{begin}}
{{onsuccess fax}}
{{fax 503-555-4489}}
Body of the document to be sent.
{{end}}

```

You can specify destinations of *delete*, *certified*, *fax*, *email*, *mime*, or *nothing* with the `{{onsuccess}}` FCL command. For detailed information on the `{{onsuccess}}` command, including syntax and examples, see [Appendix B, “FCL Commands”](#).

Printing or Faxing a Document That Cannot Be Transmitted

You have two options for establishing actions that occur when documents are not sent:

- Set defaults in the Integration Module Configuration program.
- Use the {{onerror}} FCL command in a document to override the default.

Setting the default “error” action

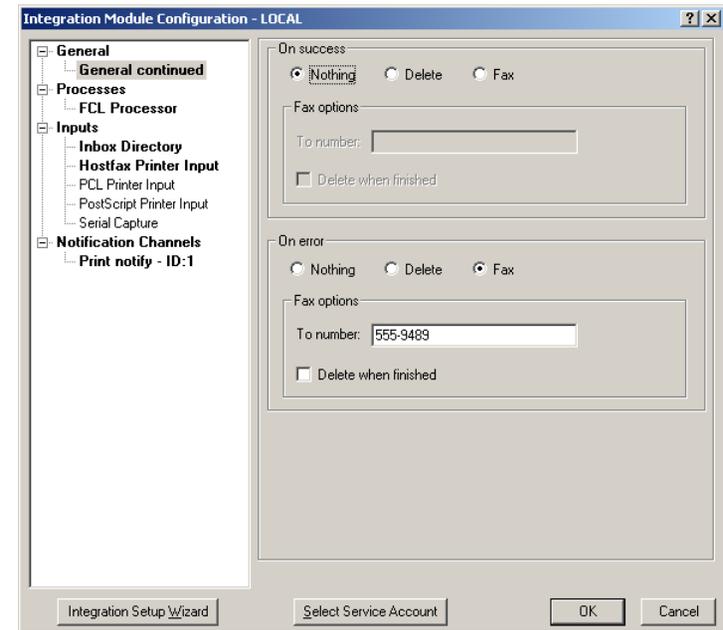
When a document cannot be successfully transmitted, a copy of the document can be sent to another recipient as a notification of the failure.

To set the error action

1. On the **Start** menu, select **Programs > RightFax > Enterprise Fax Manager**. The **Enterprise Fax Manager** window appears.
2. In the **Fax Servers** list, click the name of the server on which the Integration Module is running.
3. In the **Service Name** list, double-click **RightFax Integration Module**. The **Integration Module Configuration** window appears.

4. In the left pane, click **General continued**.

Figure 23.2 The General Continued Settings



5. Under **On error**, select one of the following options.

Table 23a On error Options

Option	Description
Nothing	No action will be taken.
Delete	The fax image will be deleted from the fax server.
Fax	Fax the document to another recipient. Enter a fax number in the To number: box. Select Delete when finished to delete the fax image from the fax server.



Note The document also can be e-mailed or sent for certified delivery using the FCL commands `{{type email}}`, `{{type certified}}`, or `{{type mime}}`. For more information, refer to [Chapter 11, “Specifying the Destination of the Document”](#).

6. Click **OK**.

Using the `{{onerror}}` command to set “error” actions for a document

To set “error” actions for a document and override the default, add the `{{onerror}}` command to the FCL. In the following example, a copy of the document will be faxed to another recipient if it is not successfully transmitted.

```

{{begin}}
{{onerror fax}}
{{fax 503-555-4489}}
Body of the document to be sent.
{{end}}
```

You can specify destinations of *delete*, *certified*, *fax*, *email*, *mime*, or *nothing* with the `{{onerror}}` command. For detailed information on the `{{onerror}}` command, including syntax and examples, see [Appendix B, “FCL Commands”](#).

Performing Actions on Documents With Missing Data

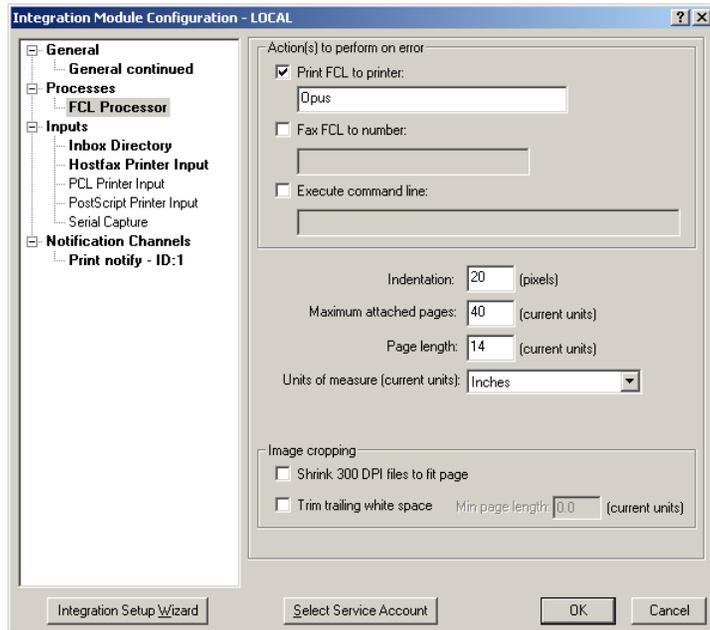
When the Integration Module processes a document, it extracts data from the FCL commands before it creates and sends the document. To notify you that a document is missing data that is required to send the document, you can send the document to another recipient in order to troubleshoot the error.

To set default actions

1. From the **Start** menu, select **Programs > RightFax > Enterprise Fax Manager**. The **Enterprise Fax Manager** window appears.
2. In the **Fax Servers** list, click the name of the server on which the Integration Module is running.
3. In the **Service Name** list, double-click **RightFax Integration Module**. The **Integration Module Configuration** window appears.

- In the left pane, click **FCL Processor**.

Figure 23.3 The FCL Processor Settings



- To print the document, select **Print FCL to printer**, and then enter the name of the printer as defined in the Enterprise Fax Manager.
- To fax the document, select **Fax FCL to number**, and enter the fax number.

- To send the document using a notification channel, select **Execute command line**, and then enter an appropriate command line. To do so, refer to the instructions for writing notification command lines in [Chapter 21, "Setting Up Notification Messages of Document Transmission"](#).

Example The document can be sent to Microsoft Exchange as an e-mail message. Therefore, in the **Execute command line** box, you enter:

```
fax2mapi -t marco.navarro@shs.com -f $$ -u "MS Exchange Settings"
```

When a document fails because data is missing, the Integration Module will send it to the e-mail address marco.navarro@shs.com.

■ ■ ■

Chapter 24

Programming for the RightFax XML Interface

The RightFax XML Interface converts XML to FCL. For host applications that produce XML and can't be modified to produce FCL, the RightFax XML Interface provides a beneficial option.

When the Integration Module receives an XML-based document, the RightFax XML Interface converts it to an FCL-encoded document.

Introduction to the RightFax XML Interface

The XML Interface software performs four functions (submit, query, action, and notification) via three methods of transport (HTTP or HTTPS, FTP, and IBM WebSphere MQ). XML Interface functionality is achieved by creating XML documents that adhere to RightFax schemas ([page 147](#)).

The RightFax API for Java is for Java programmers. It provides an alternate method of creating and sending XML to the RightFax server. The API for Java allows access to XML Interface functionality without requiring that a customer know XML or the RightFax XML Interface schemas. For more information on the API for Java, see [Chapter 25](#), “Programming for the RightFax API for Java”.

The availability of XML Interface functions depends on the method of transport.

Table 24a XML Interface Transport Methods and Functions

Action	HTTP or HTTPS	File	IBM WebSphere MQ
Action	Yes	Yes	Yes
Action Reply	Yes	Yes	No
Notification	Yes	Yes	Yes
Query	Yes	Yes	No
Query Reply	Yes	Yes	No
Submit	Yes	Yes	Yes
Submit Reply	Yes	Yes	No

Installing the XML Interface



Note RightFax provides programming interfaces for both XML and Java. These interfaces are both installed when you run the XML installation.

Minimum System Requirements

In addition to the minimum system requirements for the RightFax server and Integration Module, Microsoft Internet Information Server (IIS) version 6.0 or later must be installed on the RightFax server.

If you are installing the XML interface on a server running IIS version 6.0, **Active Server Pages** and **ISAPI Extensions** must be enabled. Parent Paths must also be enabled, either for the Default Web Site as a whole, or for the RFXML virtual directory which appears after the installation of the Java/XML API.

If you are installing the XML interface on a server running IIS version 7.0, you must enable **CGI modules** and **ISAPI modules**. Parent Paths must also be enabled, either for the Default Web Site as a whole, or for the RFXML virtual directory which appears after the installation of the Java/XML API.

Installing the XML Interface

Follow the instructions for installing the RightFax server in the *RightFax Installation Guide* and use the following specific steps:

- On the **Setup Type** screen, select **Custom** and then click **Next**.
- On the **Setup Features** screen, expand the **RightFax Server** heading in the components tree and select the **Java/XML API** component to install. Click **Next**.

After the RightFax server and Java/XML API component is installed and activated, you must configure an SMTP host and an IIS user account for the **rfxml** website.

SMTP Host. Open the RightFax Server Module and click the e-transport tab. Enter the name of the SMTP server on your network that will transport all SMTP alerts and notifications regarding the RightFax server. If you will not be using SMTP to deliver RightFax alerts and notifications, you can leave this option blank.

IIS User Account. In IIS, configure the **rfxml** website with an IIS user account that RightFax will use to access the IIS server. This account is required for Java development.

Additional XML development tools are located in the `\Program Files\RightFax\Production\XML` folder.

XML Interface Functions

Note that not every function is available for each transport method. For more information, see [“Table 24a: XML Interface Transport Methods and Functions”](#).

Submit

To send an outgoing document, use the `submit` function. This function is governed by the schema `XML_FAX_SUBMIT`, which defines the optional and required XML tags needed to submit a document to be sent by the RightFax server. The `submit` schema includes information such as fax number, contact name, owner, and attachments.

When you submit a document, you have the option of creating a unique ID for each recipient or letting the RightFax server assign a unique ID for you. For unique IDs that the software creates, the format is:

- The first seven characters are the name of the RightFax server
- The last eight characters are a number unique to a document



Note *If you create your own unique ID, it must be 15 characters or less.*

When creating an XML document, it must match the RightFax XML schema (see [“The Schemas”](#) on [page 147](#)). If the document does not match the applicable schema—out of preference, convenience, or any for other reason—then you must create an XSLT to convert the document to the RightFax schema. Or, you can create an XSLT to convert the document directly to FCL.

The companion to XML_FAX_SUBMIT is XML_FAX_SUBMIT_REPLY, which informs you of each recipient's unique ID (if you did not create your own when you submitted the document). The reply includes a message such as "Document has been submitted for sending," indicating that the first step has been completed successfully. If you configured RightFax to return a notification when the document transmission is completed, then you will also receive an XML_FAX_NOTIFICATION report on the status of the transaction that is more detailed than XML_FAX_SUBMIT_REPLY.

Query

To determine the status of any document in the RightFax system, including those not originating from XML, use the `query` function. This function is governed by the schema XML_FAX_QUERY, which defines the optional and required XML tags needed to submit a query to the RightFax server. The `query` schema includes search criteria such as date, time, and unique ID.

The companion to XML_FAX_QUERY is XML_FAX_QUERY_REPLY, which is a report informing you of the query results.

Action

To perform an action on a sent or received document, use the `action` function. This function is governed by the schema XML_FAX_ACTION, which defines the optional and required XML tags needed to perform the action. You must specify the relevant document with its unique ID, and then include instructions to delete, forward, or create a library document.

The companion to XML_FAX_ACTION is XML_FAX_ACTION_REPLY, which is a report informing you of the status of the action you requested.

Notification

To receive a notification that a document was or was not sent successfully from the RightFax server, create a notification that uses the XML_FAX_NOTIFICATION.INC template. This function is governed by the schema XML_FAX_NOTIFICATION, which defines the XML tags used to create a notification. Notifications are a one-way transaction. That is, the client cannot "notify" the server. Rather, the user includes a notification request when using the `submit` function, and the server sends a notification when the submitted document is either sent successfully or not sent successfully.

The notification function is different from the submit-reply function. The message returned with submit-reply states that the server has received the document from the client. The message returned with notification states that the server has sent (or not sent) the document to its destination.

For information on creating notifications, see [Chapter 21, "Setting Up Notification Messages of Document Transmission"](#).

Transports

Note that some transport methods do not support all of the RightFax XML Interface functions. For more information, see ["Table 24a: XML Interface Transport Methods and Functions"](#).

You can provide transport and access list security using standard mechanisms. Use HTTPS instead of HTTP to achieve transport encryption. The Web server providing the RFWebCon.dll resource can also generate user, IP address, and domain restrictions.

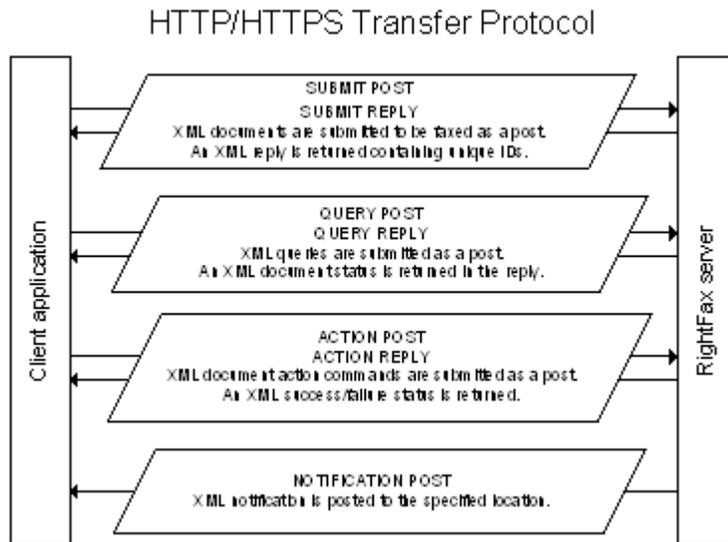
The XML Interface software supports three transport methods: HTTP or HTTPS, FTP, and IBM WebSphere MQ. You can also use the RightFax API for Java to create and send XML via HTTP (see [Chapter 25, "Programming for the RightFax API for Java"](#)).

HTTP or HTTPS transport

The HTTP/S transport sends XML to and from the client/server by executing a series of HTTP/S post methods (RFC 2616 [1]). You can perform three posts (submit, query, and action) and receive a success or failure notification from the RightFax server. All XML Interface functions are supported.

Each HTTP/S post is formatted with the library name of the Captaris connector (RFWebCon.dll) and the method to be executed stored in the X-Captaris-Method field. Valid methods are submit, query, and action. The format of the post line includes standard HTTP headers as well as XML data.

Figure 24.1 HTTP/HTTPS Transfer Protocol for XML Documents



Example HTTP/S Post

Table 24b Example HTTP/S Post

Call	Response
POST /RFWebCon.DLL HTTP/1.1 Host: www.captaris.com Content-Type: text/xml Content-Length: nnnn X-Captaris-Method: submit	HTTP/1.1 200 OK Content-Type: text/xml Content-Length: nnnn X-Captaris-Method: submit_reply <XML_FAX_SUBMIT_REPLY>... </XML_FAX_SUBMIT_REPLY>
<XML_FAX_SUBMIT>... </XML_FAX_SUBMIT>	

HTTP/S Attachments

Attachments are valid only for the submit function.

Inline

XML does not support binary data. To be included inline with the XML, binary attachments must be encoded using BASE64. BASE64 represents binary data using a 64-character subset of International Alphabet IA5. For information on implementation details of BASE64, see RFC 1421 [2].

Multipart MIME

When attachments are included inline, they significantly increase the size of the XML message. Some XML parser implementations load the entire XML message into memory, which causes performance degradation when large messages are processed.

An alternative to inline attachments is to encapsulate the XML message and attachment using multipart MIME. Specifically, multipart or mixed content type is used to separate the XML document from each attachment. The first attachment with the

text/XML media type listed for its content type is considered the XML_FAX_SUBMIT document. All other documents within the MIME message are considered attachments to be included with the message.

Replies (XML_FAX_REPLY) are never returned as multipart MIME messages. The data contained within a reply is minimal and does not include binary attachments.

HTTP or HTTPS notifications

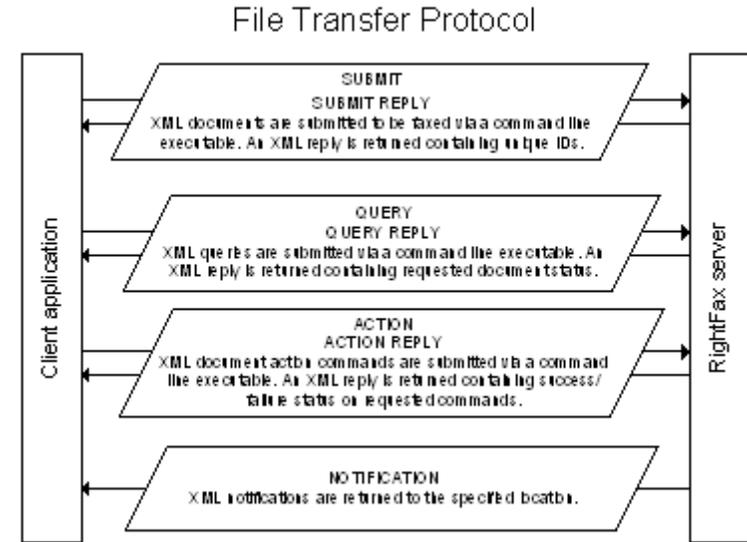
To receive notifications in this manner, you must provide an HTTP listener. The application Postfile.exe can be used to send messages from the RightFax server, but it needs the path to the HTTP listener for you to actually receive the message.

File transport

File transport for XML allows an XML document to be processed using a command line executable (Parsexml.exe). When combined with existing Captaris technology, this can be used for printer input, directory scanning, and various other input methods.

Some methods, such as printer input, do not contain a mechanism to receive a reply.

Figure 24.2 File Transfer Protocol for XML Documents



Command line syntax

parseXML [options] <xmlfile> [attachments]

Example

parseXML submit.xml po.doc list.pdf

Command line options

- **-q**—XML input is Query document. Default is Submit document.
- **-a**—XML input is Action document. Default is Submit document.
- **-f**—<filename> Output reply XML to this file.

File transport attachments

Attachments are valid only for the submit function.

Inline

XML does not support binary data. To be included inline with the XML, binary attachments must be encoded using BASE64. BASE64 represents binary data using a 64-character subset of International Alphabet IA5. For information on implementation details of BASE64, see RFC 1421 [2].

File attachments

When attachments are included inline, they significantly increase the size of the XML message. Some XML parser implementations load the entire XML message into memory, which causes performance degradation when large messages are processed. An alternative to inline attachments is to include them as separate files.

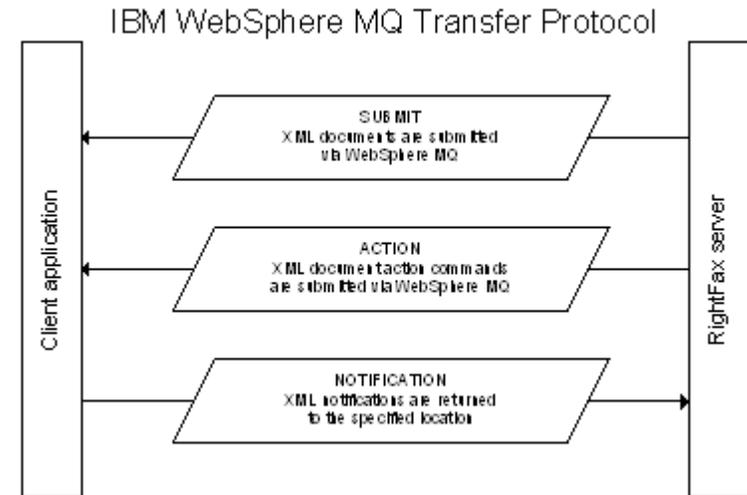
In the following example command line, po.doc and list.pdf are attached files:

```
parseXML submit.xml po.doc list.pdf
```

IBM WebSphere MQ transport

IBM WebSphere MQ acts as an intermediary in connecting the host application to the RightFax server software. It offers multiple connection options and requires some advanced configuration. For more information, see the IBM's WebSphere MQ documentation.

Figure 24.3 IBM WebSphere MQ Transfer Protocol for XML Documents



IBM WebSphere MQ attachments

Attachments are valid only for the submit function.

Inline

XML does not support binary data. To be included inline with the XML, binary attachments must be encoded using BASE64. BASE64 represents binary data using a 64-character subset of International Alphabet IA5. For information on implementation details of BASE64, see RFC 1421 [2].

Understanding Body and Cover Text

Body and cover text is relevant only for the `submit` function. The data for the body and `cover_text` nodes can be any file format. Plain text (ASCII) allows for the simplest transition to XML, but any format is acceptable. Rich Text Format (.rtf) and Microsoft Word Document (.doc) formats are examples of formatted text files.

Using plain text data

If the data is in plain text (such as .txt) format, then copy the data straight into the XML nodes, with no additional attributes needed.

Using formatted text data

If the data is not plain text format, then:

1. Use the `type` attribute, setting it to the extension associated with that document type.
2. Copy the data into the XML node.

Using binary data

If the data is binary (.doc, .pdf, etc.), then:

1. Encode the data using BASE64 encoding.
2. Use the `type` attribute, setting it to the extension associated with that document type.
3. Use the `encoding` attribute, setting it to the type of encoding.

4. Copy the encoded data into the XML node.

Example In the following example, the cover text is rich text format (.rtf) and the body text is in Microsoft Word (.doc) format.

```
...
<!--The body is a Word document, and the cover text is
RTF-->
<COVER_TEXT type="RTF">
{\rtf1\ansi\ansicpg1252\uc1 ..... \par }}
</COVER_TEXT>
<BODY type="DOC" encoding="BASE64">
0M8R4KGx.....AAAAAAAAA=
</BODY>
```

The Schemas

The schemas listed in this section define the structure and type of content that the RightFax server can accept in XML documents. When creating an XML document, it must match the applicable schema described in this section. If the document does not match the applicable schema, then you must create an XSLT to convert the document to the RightFax schema. Or, you can create an XSLT to convert the document directly to FCL. For sample documents, [“Sample Documents Based on the Schemas” on page 157](#).

XML_FAX_SUBMIT_schema

For an example document based on this schema, "[XML_FAX_SUBMIT](#)" on [page 157](#).

```
<?xml version="1.0" encoding="UTF-8"?>
<!--rfxml version="2.0" -->
  <Schema name="XML_FAX_SUBMIT" xmlns="urn:schemas-microsoft-com:xml-data" xmlns:dt="urn:schemas-microsoft-com:datatypes">
    <ElementType name="SEND_DATE_TIME" model="closed" content="textOnly" dt:type="dateTime.tz"/>
    <ElementType name="INCLUDE_BEG" model="closed" content="textOnly" dt:type="string"/>
    <ElementType name="ADD_LIBDOC" model="closed" content="textOnly" dt:type="string"/>
    <ElementType name="INCLUDE_END" model="closed" content="textOnly" dt:type="string"/>
  <!--SENDER element-->
    <ElementType name="FROM_NAME" model="closed" content="textOnly" dt:type="string"/>
    <ElementType name="EMP_ID" model="closed" content="textOnly" dt:type="string"/>
    <ElementType name="FROM_COMPANY" model="closed" content="textOnly" dt:type="string"/>
    <ElementType name="FROM_DEPARTMENT" model="closed" content="textOnly" dt:type="string"/>
    <ElementType name="FROM_PHONE" model="closed" content="textOnly" dt:type="string"/>
    <ElementType name="RETURN_EMAIL" model="closed" content="textOnly" dt:type="string"/>
    <ElementType name="BILLINFO1" model="closed" content="textOnly" dt:type="string"/>
    <ElementType name="BILLINFO2" model="closed" content="textOnly" dt:type="string"/>
    <ElementType name="REPLY_TO" model="closed" content="textOnly" dt:type="uri"/>
    <ElementType name="RF_USER" model="closed" content="textOnly" dt:type="string"/>
    <ElementType name="SENDER" model="closed" content="eltOnly" order="seq">
      <element type="FROM_NAME" minOccurs="0" maxOccurs="1"/>
      <element type="EMP_ID" minOccurs="0" maxOccurs="1"/>
      <element type="FROM_COMPANY" minOccurs="0" maxOccurs="1"/>
      <element type="FROM_DEPARTMENT" minOccurs="0" maxOccurs="1"/>
      <element type="FROM_PHONE" minOccurs="0" maxOccurs="1"/>
      <element type="RETURN_EMAIL" minOccurs="0" maxOccurs="1"/>
      <element type="BILLINFO1" minOccurs="0" maxOccurs="1"/>
      <element type="BILLINFO2" minOccurs="0" maxOccurs="1"/>
      <element type="REPLY_TO" minOccurs="0" maxOccurs="1"/>
      <element type="RF_USER" minOccurs="1" maxOccurs="1"/>
    </ElementType>
  <!--DESTINATION STUFF-->
  <!--Common Recipient elements and attributes-->
    <AttributeType name="unique_id" dt:type="string"/>
    <ElementType name="INCLUDE_INC" model="closed" content="textOnly" dt:type="string"/>
```

```

<ElementType name="TO_NAME" model="closed" content="textOnly" dt:type="string"/>
<ElementType name="INCLUDE_DEF" model="closed" content="textOnly" dt:type="string"/>
<ElementType name="TO_COMPANY" model="closed" content="textOnly" dt:type="string"/>
<ElementType name="TO_CONTACTNUM" model="closed" content="textOnly" dt:type="string"/>
<ElementType name="NOTIFY_HOST" model="closed" content="empty">
<AttributeType name="SuccessTemplate" dt:type="string" required="yes"/>
<AttributeType name="FailureTemplate" dt:type="string"/>
<AttributeType name="Name" dt:type="string" required="yes"/>
  <attribute type="SuccessTemplate"/>
  <attribute type="FailureTemplate"/>
  <attribute type="Name"/>
</ElementType>
<ElementType name="COVERSHEET" model="closed" content="textOnly" dt:type="string"/>
<!--FAX element-->
<ElementType name="ALT_FAX_NUM" model="closed" content="textOnly" dt:type="string"/>
<ElementType name="TO_FAXNUM" model="closed" content="textOnly" dt:type="string"/>
<ElementType name="FAX" model="closed" content="eltOnly" order="seq">
  <attribute type="unique_id"/>
  <element type="TO_FAXNUM" minOccurs="1" maxOccurs="1"/>
  <element type="INCLUDE_INC" minOccurs="0" maxOccurs="1"/>
  <element type="TO_NAME" minOccurs="0" maxOccurs="1"/>
  <element type="TO_COMPANY" minOccurs="0" maxOccurs="1"/>
  <element type="ALT_FAX_NUM" minOccurs="0" maxOccurs="1"/>
  <element type="TO_CONTACTNUM" minOccurs="0" maxOccurs="1"/>
  <element type="NOTIFY_HOST" minOccurs="0" maxOccurs="1"/>
  <element type="COVERSHEET" minOccurs="0" maxOccurs="1"/>
  <element type="INCLUDE_DEF" minOccurs="0" maxOccurs="1"/>
</ElementType>
<!--EMAIL element-->
<ElementType name="SUBJECT" model="closed" content="textOnly" dt:type="string"/>
<ElementType name="CC_EMAIL" model="closed" content="textOnly" dt:type="string"/>
<ElementType name="BCC_EMAIL" model="closed" content="textOnly" dt:type="string"/>
<ElementType name="TO_EMAIL" model="closed" content="textOnly" dt:type="string"/>
<ElementType name="EMAIL" model="closed" content="eltOnly" order="seq">
  <attribute type="unique_id"/>
  <element type="TO_EMAIL" minOccurs="1" maxOccurs="1"/>
  <element type="INCLUDE_INC" minOccurs="0" maxOccurs="1"/>

```

```

    <element type="TO_NAME" minOccurs="0" maxOccurs="1"/>
    <element type="TO_COMPANY" minOccurs="0" maxOccurs="1"/>
    <element type="TO_CONTACTNUM" minOccurs="0" maxOccurs="1"/>
    <element type="NOTIFY_HOST" minOccurs="0" maxOccurs="1"/>
    <element type="COVERSHEET" minOccurs="0" maxOccurs="1"/>
    <element type="INCLUDE_DEF" minOccurs="0" maxOccurs="1"/>
    <element type="SUBJECT" minOccurs="0" maxOccurs="1"/>
    <element type="CC_EMAIL" minOccurs="0" maxOccurs="1"/>
    <element type="BCC_EMAIL" minOccurs="0" maxOccurs="1"/>
  </ElementType>
<!--PRINT element-->
  <ElementType name="COPIES" model="closed" content="textOnly" dt:type="i1"/>
  <ElementType name="PRINTER_NAME" model="closed" content="textOnly" dt:type="string"/>
  <ElementType name="PRINT" model="closed" content="eltOnly" order="seq">
    <attribute type="unique_id"/>
    <element type="COPIES" minOccurs="0" maxOccurs="1"/>
    <element type="PRINTER_NAME" minOccurs="0" maxOccurs="1"/>
    <element type="INCLUDE_INC" minOccurs="0" maxOccurs="1"/>
    <element type="TO_NAME" minOccurs="0" maxOccurs="1"/>
    <element type="TO_COMPANY" minOccurs="0" maxOccurs="1"/>
    <element type="TO_CONTACTNUM" minOccurs="0" maxOccurs="1"/>
    <element type="COVERSHEET" minOccurs="0" maxOccurs="1"/>
    <element type="INCLUDE_DEF" minOccurs="0" maxOccurs="1"/>
  </ElementType>
  <ElementType name="DESTINATIONS" model="closed" content="eltOnly" order="seq">
    <element type="FAX" minOccurs="0" maxOccurs="*/>
    <element type="EMAIL" minOccurs="0" maxOccurs="*/>
    <element type="PRINT" minOccurs="0" maxOccurs="*/>
  </ElementType>
<!--END DESTINATION STUFF-->
<!--FORM element-->
  <ElementType name="FORM" model="closed" content="textOnly" dt:type="string">
    <AttributeType name="xcoord" dt:type="number"/>
    <AttributeType name="ycoord" dt:type="number"/>
    <attribute type="xcoord"/>
    <attribute type="ycoord"/>
  </ElementType>

```

```

<!--ADD_IMAGE element-->
  <ElementType name="ADD_IMAGE" model="closed" content="textOnly" dt:type="string">
    <AttributeType name="page" dt:type="enumeration" dt:values="CURRENT ALL LAST"/>
    <AttributeType name="xoffset" dt:type="i1"/><AttributeType name="yoffset" dt:type="i1"/>
      <attribute type="page" default="CURRENT"/>
      <attribute type="xoffset"/>
      <attribute type="yoffset"/>
    </ElementType>
<!--BODY and COVER_TEXT common attributes-->
  <AttributeType name="type" dt:type="string"/>
  <AttributeType name="encoding" dt:type="enumeration" dt:values="NONE BASE64 QUOTEDPRINTABLE"/>
<!--COVER_TEXT element-->
  <ElementType name="COVER_TEXT" model="closed" content="textOnly" dt:type="string">
    <attribute type="type" default="TXT"/>
    <attribute type="encoding"/>
  </ElementType>
<!--BODY element-->
  <ElementType name="BODY" model="closed" content="mixed">
    <element type="FONT" minOccurs="0" maxOccurs="1"/>
    <AttributeType name="tm" dt:type="i1"/>
    <AttributeType name="lm" dt:type="i1"/>
    <AttributeType name="bm" dt:type="i1"/>
    <AttributeType name="font_name" dt:type="string"/>
    <AttributeType name="font_leading" dt:type="i1"/>
    <AttributeType name="font_pitch" dt:type="i1"/>
      <attribute type="type" default="TXT"/>
      <attribute type="encoding"/>
      <attribute type="tm"/>
      <attribute type="lm"/>
      <attribute type="bm"/>
      <attribute type="font_name"/>
      <attribute type="font_leading"/>
      <attribute type="font_pitch"/>
    </ElementType>
<!--ATTACHMENT element-->
  <ElementType name="ATTACHMENT" model="closed" content="eltOnly" order="one">
    <element type="DATA"/>

```

```

    <element type="FILE"/>
  </ElementType>
  <ElementType name="DATA" model="closed" content="textOnly" dt:type="string">
    <attribute type="type" default="TXT"/>
    <attribute type="encoding" default="NONE"/>
  </ElementType>
  <ElementType name="FILE" model="closed" content="empty">
    <AttributeType name="path" dt:type="string" required="yes"/>
    <AttributeType name="delete" dt:type="string" required="no"/>
    <attribute type="path"/>
    <attribute type="delete"/>
  </ElementType>
<!--Root element-->
  <ElementType name="XML_FAX_SUBMIT" model="closed" content="eltOnly" order="seq">
    <AttributeType name="java" dt:type="boolean"/>
    <AttributeType name="stylesheet" dt:type="string"/>
    <attribute type="java" default="0"/>
    <attribute type="stylesheet" default="XML_FAX_SUBMIT.XSL"/>
    <element type="SEND_DATE_TIME" minOccurs="0" maxOccurs="1"/>
    <element type="INCLUDE_BEG" minOccurs="0" maxOccurs="1"/>
    <element type="SENDER" minOccurs="0" maxOccurs="1"/>
    <element type="DESTINATIONS" minOccurs="1" maxOccurs="1"/>
    <element type="FORM" minOccurs="0" maxOccurs="1"/>
    <element type="ADD_IMAGE" minOccurs="0" maxOccurs="1"/>
    <element type="COVER_TEXT" minOccurs="0" maxOccurs="1"/>
    <element type="BODY" minOccurs="0" maxOccurs="1"/>
    <element type="ATTACHMENT" minOccurs="0" maxOccurs="*/>
    <element type="ADD_LIBDOC" minOccurs="0" maxOccurs="1"/>
    <element type="INCLUDE_END" minOccurs="0" maxOccurs="1"/>
  </ElementType>
</Schema>

```

XML_FAX_SUBMIT_REPLY.dtd

For an example document based on this dtd, see [“XML_FAX_SUBMIT_REPLY”](#) on page 160.

```
<!ELEMENT XML_FAX_SUBMIT_REPLY (FAX+)>
<!ATTLIST FAX unique_id CDATA #REQUIRED>
<!ELEMENT FAX (RETURN_CODE, MESSAGE)>
<!ELEMENT RETURN_CODE (#PCDATA)>
<!ELEMENT MESSAGE (#PCDATA)>
```

XML_FAX_QUERY_schema

For an example document based on this schema, see [“XML_FAX_QUERY”](#) on page 160.

```
<?xml version="1.0"?>
<Schema xmlns="urn:schemas-microsoft-com:xml-data" xmlns:dt="urn:schemas-microsoft-com:datatypes">
  <ElementType name="UNIQUE_ID" content="textOnly"/>
  <AttributeType name="start" dt:type="datetime.tz" required="yes"/>
  <AttributeType name="end" dt:type="datetime.tz" required="yes"/>
  <ElementType name="DATE_RANGE" content="textOnly">
    <attribute type="start"/>
    <attribute type="end"/>
  </ElementType>
  <AttributeType name="faxtype" dt:type="enumeration" dt:values="OUTBOUND INBOUND BOTH"/>
  <ElementType name="TO_FAXNUM" content="textOnly"/>
  <ElementType name="RF_USER" content="textOnly"/>
  <ElementType name="STATUS" content="textOnly"/>
  <ElementType name="QUERY" content="eltOnly">
    <attribute type="faxtype" default="OUTBOUND"/>
    <element type="UNIQUE_ID" minOccurs="0" maxOccurs="1"/>
    <element type="DATE_RANGE" minOccurs="0" maxOccurs="1"/>
    <element type="TO_FAXNUM" minOccurs="0" maxOccurs="1"/>
    <element type="RF_USER" minOccurs="0" maxOccurs="1"/>
    <element type="STATUS" minOccurs="0" maxOccurs="1"/>
  </ElementType>
  <ElementType name="QUERIES" content="eltOnly">
    <element type="QUERY" minOccurs="1" maxOccurs="*" />
  </ElementType>
  <ElementType name="XML_FAX_QUERY" content="eltOnly">
```

```

    <element type="QUERIES"/>
  </ElementType>
</Schema>

```

XML_FAX_QUERY_REPLY.dtd

For an example document based on this dtd, see ["XML_FAX_QUERY_REPLY"](#) on [page 161](#).

```

<!ELEMENT XML_FAX_QUERY_REPLY (MESSAGE?, FAXSTATUS*)>
<!ELEMENT MESSAGE (#PCDATA)>
<!ELEMENT FAXSTATUS (STATUS_CODE, STATUS_MSG, ERROR_CODE, DISPOSITION, TERMSTAT, OWNER_ID, TO_FAXNUM,
  TO_CONTACTNUM, TO_NAME, TO_COMPANY, TO_CITYSTATE, FROM_NAME, FROM_PHONENUM, BILLINFO1, BILLINFO2,
  CREATE_DATETIME, SENDTIME, REMOTEID, SEND_DATETIME, SEND_CHANNEL, CUSTOM1)>
<!ATTLIST FAXSTATUS unique_id CDATA #REQUIRED>
<!ATTLIST FAXSTATUS query_id CDATA #IMPLIED>
<!ELEMENT STATUS_CODE (#PCDATA)>
<!ELEMENT STATUS_MSG (#PCDATA)>
<!ELEMENT ERROR_CODE (#PCDATA)>
<!ELEMENT DISPOSITION (#PCDATA)>
<!ELEMENT TERMSTAT (#PCDATA)>
<!ELEMENT OWNER_ID (#PCDATA)>
<!ELEMENT TO_FAXNUM (#PCDATA)>
<!ELEMENT TO_CONTACTNUM (#PCDATA)>
<!ELEMENT TO_NAME (#PCDATA)>
<!ELEMENT TO_COMPANY (#PCDATA)>
<!ELEMENT TO_CITYSTATE (#PCDATA)>
<!ELEMENT FROM_NAME (#PCDATA)>
<!ELEMENT FROM_PHONENUM (#PCDATA)>
<!ELEMENT BILLINFO1 (#PCDATA)>
<!ELEMENT BILLINFO2 (#PCDATA)>
<!ELEMENT CREATE_DATETIME (#PCDATA)>
<!ELEMENT SENDTIME (#PCDATA)>
<!ELEMENT REMOTEID (#PCDATA)>
<!ELEMENT SEND_DATETIME (#PCDATA)>
<!ELEMENT SEND_CHANNEL (#PCDATA)>
<!ELEMENT CUSTOM1 (#PCDATA)>

```

XML_FAX_ACTION_schema

For an example document based on this schema, see ["XML_FAX_ACTION"](#) on page 162.

```
<?xml version="1.0"?>
<Schema xmlns="urn:schemas-microsoft-com:xml-data" xmlns:dt="urn:schemas-microsoft-com:datatypes">
  <ElementType name="TO_NAME" content="textOnly"/>
  <ElementType name="TO_COMPANY" content="textOnly"/>
  <ElementType name="ALT_FAX_NUM" content="textOnly"/>
  <ElementType name="TO_CONTACTNUM" content="textOnly"/>
  <ElementType name="COVERSHEET" content="textOnly"/>
  <ElementType name="TO_FAXNUM" content="textOnly"/>
  <ElementType name="FAX_RECIPIENT" content="eltOnly">
    <element type="TO_NAME" minOccurs="0" maxOccurs="1"/>
    <element type="TO_COMPANY" minOccurs="0" maxOccurs="1"/>
    <element type="ALT_FAX_NUM" minOccurs="0" maxOccurs="1"/>
    <element type="TO_CONTACTNUM" minOccurs="0" maxOccurs="1"/>
    <element type="COVERSHEET" minOccurs="0" maxOccurs="1"/>
    <element type="TO_FAXNUM" />
  </ElementType>
  <ElementType name="ID" content="textOnly"/>
  <ElementType name="DESCRIPTION" content="textOnly"/>
  <ElementType name="DELETE" content="empty"/>
  <ElementType name="FORWARD" content="eltOnly">
    <element type="FAX_RECIPIENT" minOccurs="1" maxOccurs="*/>
  </ElementType>
  <ElementType name="CREATE_LIB_DOC" content="eltOnly">
    <element type="ID" minOccurs="1" maxOccurs="1"/>
    <element type="DESCRIPTION" minOccurs="1" maxOccurs="1"/>
  </ElementType>
  <AttributeType name="unique_id" dt:type="string" required="yes"/>
<!--Had to make this "closed" and set order to "one" so that only one or the other of child elements can occur-->
  <ElementType name="FAX" model="closed" content="eltOnly" order="one">
    <attribute type="unique_id"/>
    <element type="DELETE"/>
    <element type="FORWARD"/>
    <element type="CREATE_LIB_DOC"/>
  </ElementType>
</Schema>
```

```

<AttributeType name="docid" dt:type="string"/>
<ElementType name="XML_FAX_ACTION" content="eltOnly">
  <element type="FAX" minOccurs="1" maxOccurs="1"/>
</ElementType>
</Schema>

```

XML_FAX_ACTION_REPLY.dtd

For an example document based on this dtd, see ["XML_FAX_ACTION_REPLY"](#) on page 162.

```

<!ELEMENT XML_FAX_ACTION (FAX+)>
<!ELEMENT FAX (DELETE | FORWARD)>
<!ATTLIST FAX unique_id CDATA #REQUIRED>
<!ELEMENT DELETE EMPTY>
<!ELEMENT FORWARD FAX_RECIPIENT+>
<!ELEMENT FAX_RECIPIENT (TO_NAME?, TO_COMPANY?, ALT_FAX_NUM?, TO_CONTACTNUM?, COVERSHEET?, TO_FAXNUM)>
<!ELEMENT TO_NAME (#PCDATA)>
<!ELEMENT TO_COMPANY (#PCDATA)>
<!ELEMENT ALT_FAX_NUM (#PCDATA)>
<!ELEMENT TO_CONTACTNUM (#PCDATA)>
<!ELEMENT COVERSHEET (#PCDATA)>
<!ELEMENT TO_FAXNUM (#PCDATA)>

```

XML_FAX_NOTIFICATION.dtd

For an example document based on this dtd, see ["XML_FAX_NOTIFICATION"](#) on page 163.

```

<!ELEMENT XML_FAX_NOTIFICATION (FAXSTATUS*)>
<!ELEMENT FAXSTATUS (STATUS_CODE, STATUS_MSG, ERROR_CODE, DISPOSITION, TERMSTAT, OWNER_ID, TO_FAXNUM,
  TO_CONTACTNUM, TO_NAME, TO_COMPANY, TO_CITYSTATE, FROM_NAME, FROM_PHONENUM, BILLINFO1, BILLINFO2,
  CREATE_DATETIME, SENDTIME, REMOTEID, SEND_DATETIME, SEND_CHANNEL, CUSTOM1)>
<!ATTLIST FAXSTATUS unique_id CDATA #REQUIRED>
<!ELEMENT STATUS_CODE (#PCDATA)>
<!ELEMENT STATUS_MSG (#PCDATA)>
<!ELEMENT ERROR_CODE (#PCDATA)>
<!ELEMENT DISPOSITION (#PCDATA)>
<!ELEMENT TERMSTAT (#PCDATA)>
<!ELEMENT OWNER_ID (#PCDATA)>
<!ELEMENT TO_FAXNUM (#PCDATA)>

```

```

<!ELEMENT TO_CONTACTNUM (#PCDATA)>
<!ELEMENT TO_NAME (#PCDATA)>
<!ELEMENT TO_COMPANY (#PCDATA)>
<!ELEMENT TO_CITYSTATE (#PCDATA)>
<!ELEMENT FROM_NAME (#PCDATA)>
<!ELEMENT FROM_PHONENUM (#PCDATA)>
<!ELEMENT BILLINFO1 (#PCDATA)>
<!ELEMENT BILLINFO2 (#PCDATA)>
<!ELEMENT CREATE_DATETIME (#PCDATA)>
<!ELEMENT SENDTIME (#PCDATA)>
<!ELEMENT REMOTEID (#PCDATA)>
<!ELEMENT SEND_DATETIME (#PCDATA)>
<!ELEMENT SEND_CHANNEL (#PCDATA)>
<!ELEMENT CUSTOM1 (#PCDATA)>

```

Sample Documents Based on the Schemas

This section lists sample documents based on the XML Interface schemas and corresponding DTDs. For the schemas and DTDs, see [“The Schemas”](#) on [page 147](#).



Warning For example documents that require a name space (e.g., `<XML_FAX_SUBMIT stylesheet="XML_FAX_SUBMIT.XSL" xmlns="namespace">`), the name space you write must exactly match the corresponding name space in the XSLT file. If it does not, the FCL documents that appear in the inbox will be empty.

XML_FAX_SUBMIT

```

<?xml version="1.0"?>
<!--rfxml version="2.0" -->
<XML_FAX_SUBMIT stylesheet="XML_FAX_SUBMIT.XSL" xmlns="x-schema:c:\program files\rightfax\production\xml
\schemas\XML_FAX_SUBMIT_schema.xml">
  <SEND_DATE_TIME>2000-01-24T15:20:00-08:00</SEND_DATE_TIME>
  <INCLUDE_BEG>xml.beg</INCLUDE_BEG>
  <SENDER>
    <FROM_NAME>Bob McKenzie</FROM_NAME>

```

```

<EMP_ID>555-66-7777</EMP_ID>
<FROM_COMPANY>Company, Ltd.</FROM_COMPANY>
<FROM_DEPARTMENT>Store</FROM_DEPARTMENT>
<FROM_PHONE>555-9876</FROM_PHONE>
<RETURN_EMAIL>bobm@company.com</RETURN_EMAIL>
<BILLINFO1>12345</BILLINFO1>
<BILLINFO2>678dk</BILLINFO2>
<REPLY_TO>http://www.company.com/faxreply</REPLY_TO>
<RF_USER>bobm</RF_USER>
</SENDER>
<DESTINATIONS>
  <FAX unique_id="PRODXML:0001">
    <TO_FAXNUM>555-1111</TO_FAXNUM>
    <INCLUDE_INC>xml.inc</INCLUDE_INC>
    <TO_NAME>Fred Flintstone</TO_NAME>
    <TO_COMPANY>Acme, Inc.</TO_COMPANY>
    <ALT_FAX_NUM>555-1112</ALT_FAX_NUM>
    <TO_CONTACTNUM>555-6543</TO_CONTACTNUM>
    <NOTIFY_HOST SuccessTemplate="XML_FAX_NOTIFICATION.inc" FailureTemplate="XML_FAX_NOTIFICATION.inc"
      Name="XMLNotify"/>
    <COVERSHEET>auto.cov</COVERSHEET>
    <INCLUDE_DEF>xml.def</INCLUDE_DEF>
  </FAX>
  <FAX unique_id="PRODXML:0002">
    <TO_FAXNUM>555-1234</TO_FAXNUM>
    <TO_NAME>Bill Smith</TO_NAME>
  </FAX>
  <EMAIL unique_id="PRODXML:0003">
    <TO_EMAIL>barney@captaris.com</TO_EMAIL>
    <TO_NAME>Barney Smith </TO_NAME>
    <NOTIFY_HOST SuccessTemplate="exmlsuc.inc" FailureTemplate="exmlerr.inc" Name="XMLNotif"/>
    <SUBJECT>Here is my fax </SUBJECT>
    <CC_EMAIL>fred@captaris.com</CC_EMAIL>
    <BCC_EMAIL>wilma@captaris.com</BCC_EMAIL>
  </EMAIL>
  <PRINT unique_id="PRODXML:0003">
    <COPIES>2</COPIES>

```

```

    <PRINTER_NAME>Dev1</PRINTER_NAME>
    <INCLUDE_INC>xml.inc</INCLUDE_INC>
    <TO_NAME>Bill Smith</TO_NAME>
    <TO_COMPANY>Acme, Inc.</TO_COMPANY>
    <TO_CONTACTNUM>555-6545</TO_CONTACTNUM>
    <COVERSHEET>auto.cov</COVERSHEET>
    <INCLUDE_DEF>xml.def</INCLUDE_DEF>
  </PRINT>
</DESTINATIONS>
<FORM xcoord="0.5" ycoord="1.0">form.inc</FORM>
<ADD_IMAGE page="CURRENT" xoffset="2" yoffset="4">simpsons2.tif</ADD_IMAGE>
<COVER_TEXT type="TXT" encoding="NONE">
  Put me on the cover!!!!
  &lt; &gt;
  Here is another line of cover text.
</COVER_TEXT>
<BODY type="TXT" encoding="NONE" tm="1" lm="1" bm="1" font_name="Arial" font_leading="12" font_pitch="12">
  Yada Yada Yada
  &lt; &gt;
  More More More
</BODY>
<ATTACHMENT>
  <DATA type="RTF" encoding="NONE">%RTF%/Arial%Put this text into the document at an attachment.%EOF%</DATA>
</ATTACHMENT>
<ATTACHMENT>
  <FILE path="eagle.tif" delete="yes"/>
</ATTACHMENT>
<ATTACHMENT>
  <FILE path="eagle2.tif"/>
</ATTACHMENT>
<ADD_LIBDOC>TANDC1</ADD_LIBDOC>
<INCLUDE_END>xml.end</INCLUDE_END>
</XML_FAX_SUBMIT>

```

XML_FAX_SUBMIT_REPLY

```

<?xml version="1.0" ?>
<XML_FAX_SUBMIT_REPLY>
  <FAX unique_id="PRODXML:2055">
    <RETURN_CODE>1</RETURN_CODE>
    <MESSAGE>Document has been successfully submitted for sending.</MESSAGE>
  </FAX>
</XML_FAX_SUBMIT_REPLY>

```

XML_FAX_QUERY

```

<?xml version="1.0" ?>
<XML_FAX_QUERY xmlns="x-schema:../schemas/XML_FAX_QUERY_schema.xml">
  <QUERIES>
    <QUERY faxtype="INBOUND">
      <UNIQUE_ID>PRODXML:0001</UNIQUE_ID>
      <DATE_RANGE start="2000-03-08T18:39:09-08:00" end="2000-03-10T18:39:09-08:00"/>
      <TO_FAXNUM>555-1111</TO_FAXNUM>
      <RF_USER>bobm</RF_USER>
      <STATUS>0</STATUS>
    </QUERY>
    <QUERY faxtype="OUTBOUND">
      <UNIQUE_ID>PRODXML:0002</UNIQUE_ID>
      <DATE_RANGE start="2000-03-08T18:39:09-08:00" end="2000-03-10T18:39:09-08:00"/>
      <TO_FAXNUM>555-1112</TO_FAXNUM>
      <RF_USER>dougm</RF_USER>
      <STATUS>0</STATUS>
    </QUERY>
    <QUERY faxtype="BOTH">
      <UNIQUE_ID>PRODXML:0003</UNIQUE_ID>
      <DATE_RANGE start="2000-03-08T18:39:09-08:00"
end="2000-03-10T18:39:09-08:00"/>
      <TO_FAXNUM>555-1113</TO_FAXNUM>
      <RF_USER>dougm</RF_USER>
      <STATUS>0</STATUS>
    </QUERY>
  </QUERIES>
</XML_FAX_QUERY>

```

```
</QUERY>
</QUERIES>
</XML_FAX_QUERY>
```

XML_FAX_QUERY_REPLY

```
<?xml version="1.0" ?>
<XML_FAX_QUERY_REPLY>
  <FAXSTATUS unique_id="PRODXML:2055" query_id="1">
    <STATUS_CODE>6</STATUS_CODE>
    <STATUS_MSG>OK</STATUS_MSG>
    <ERROR_CODE>0</ERROR_CODE>
    <DISPOSITION>0</DISPOSITION>
    <TERMSTAT>32</TERMSTAT>
    <OWNER_ID>XMLACCOUNT</OWNER_ID>
    <TO_FAXNUM>5039689601</TO_FAXNUM>
    <TO_CONTACTNUM />
    <TO_NAME>Unknown</TO_NAME>
    <TO_COMPANY />
    <TO_CITYSTATE />
    <FROM_NAME />
    <FROM_PHONENUM>(999) 999-9999</FROM_PHONENUM>
    <BILLINFO1 />
    <BILLINFO2 />
    <CREATE_DATETIME>957444538</CREATE_DATETIME>
    <SENDTIME>30</SENDTIME>
    <REMOTEID>TEST computer</REMOTEID>
    <SEND_DATETIME>957444538</SEND_DATETIME>
    <SEND_CHANNEL>0</SEND_CHANNEL>
    <CUSTOM1>0</CUSTOM1>
  </FAXSTATUS>
</XML_FAX_QUERY_REPLY>
```

XML_FAX_ACTION

```

<?xml version="1.0" ?>
<XML_FAX_ACTION xmlns="x-schema:../schemas/XML_FAX_ACTION_schema.xml">
  <FAX unique_id="PRODXML:0001">
    <DELETE/>
  </FAX>
  <FAX unique_id="PRODXML:0002">
    <FORWARD>
      <FAX_RECIPIENT>
        <TO_NAME>Fred Flintstone</TO_NAME>
        <TO_COMPANY>Acme, Inc.</TO_COMPANY>
        <ALT_FAX_NUM>555-1112</ALT_FAX_NUM>
        <TO_CONTACTNUM>555-6543</TO_CONTACTNUM>
        <COVERSHEET>auto.cov</COVERSHEET>
        <TO_FAXNUM>555-1111</TO_FAXNUM>
      </FAX_RECIPIENT>
    </FORWARD>
  </FAX>
  <FAX unique_id="PRODXML:0003">
    <CREATE_LIB_DOC>
      <ID>TANDC</ID>
      <DESCRIPTION>This is a terms and conditions document.</DESCRIPTION>
    </CREATE_LIB_DOC>
  </FAX>
</XML_FAX_ACTION>

```

XML_FAX_ACTION_REPLY

```

<?xml version="1.0" ?>
<XML_FAX_ACTION_REPLY>
  <ACTION_STATUS faxid="PRODXML:2055">
    <RETURN_CODE>1</RETURN_CODE>
    <STATUS>Fax (PRODXML:2055) had been submitted for forwarding to: Acme Chemicals</STATUS>
  </ACTION_STATUS>
</XML_FAX_ACTION_REPLY>

```

XML_FAX_NOTIFICATION

```
<?xml version="1.0" ?>
<!DOCTYPE XML_FAX_NOTIFICATION SYSTEM "dtds/XML_FAX_NOTIFICATION.dtd">
<XML_FAX_NOTIFICATION>
  <FAXSTATUS unique_id="PRODXML:0001" query_id="1">
    <STATUS_CODE>0</STATUS_CODE>
    <STATUS_MSG>It is fine</STATUS_MSG>
    <ERROR_CODE></ERROR_CODE>
    <DISPOSITION></DISPOSITION>
    <TERMSTAT></TERMSTAT>
    <OWNER_ID></OWNER_ID>
    <TO_FAXNUM></TO_FAXNUM>
    <TO_CONTACTNUM></TO_CONTACTNUM>
    <TO_NAME></TO_NAME>
    <TO_COMPANY></TO_COMPANY>
    <TO_CITYSTATE></TO_CITYSTATE>
    <FROM_NAME></FROM_NAME>
    <FROM_PHONENUM></FROM_PHONENUM>
    <BILLINFO1></BILLINFO1>
    <BILLINFO2></BILLINFO2>
    <CREATE_DATETIME></CREATE_DATETIME>
    <SENDTIME></SENDTIME>
    <REMOTEID></REMOTEID>
    <SEND_DATETIME></SEND_DATETIME>
    <SEND_CHANNEL></SEND_CHANNEL>
    <CUSTOM1></CUSTOM1>
  </FAXSTATUS>
</XML_FAX_NOTIFICATION>
```

■ ■ ■

Chapter 25

Programming for the RightFax API for Java

The RightFax API for Java generates XML and sends it to the RightFax server.

The RightFax API for Java converts Java to XML on the host computer before transmitting it to the Integration Module. Once received, the RightFax XML Interface converts the XML to FCL. The Integration Module can then process and send the document from the RightFax server.

This chapter is written for experienced Java developers. It requires the Sun Microsystems, Inc. Java Developer's Kit version 1.1.8 or later.

The RightFax API for Java can submit an outbound document, query the RightFax server for the status of a document, and perform actions (forward, delete, or create a library document) on previously sent documents.

The following XML Interface functions are available with the API for Java:

- Action
- Action Reply
- Notification
- Query
- Query Reply
- Submit
- Submit Reply

The API for Java provides access to the following functions and methods of transport in the RightFax XML Interface.

Table 25a XML Interface Transport Methods and Functions

Function	HTTP or HTTPS	File	IBM WebSphere MQ
Action	Yes	Yes	Yes
Action Reply	Yes	Yes	No
Notification	Yes	Yes	Yes
Query	Yes	Yes	No
Query Reply	Yes	Yes	No
Submit	Yes	Yes	Yes
Submit Reply	Yes	Yes	No

For information on using the RightFax API for Java, see the RightFax documentation that is installed with the API. The documentation is installed in the folder RightFax\Production\XML\Java\Docs.

Installing the API for Java



Note *RightFax provides programming interfaces for both Java and XML. These interfaces are both installed when you run the Java installation.*

Minimum System Requirements

In addition to the minimum system requirements for the RightFax server and Integration Module, Microsoft Internet Information Server (IIS) version 6.0 or later must be installed on the RightFax server.

If you are installing the XML interface on a server running IIS version 6.0, **Active Server Pages** and **ISAPI Extensions** must be enabled. Parent Paths must also be enabled, either for the Default Web Site as a whole, or for the RFXML virtual directory which appears after the installation of the Java/XML API.

If you are installing the XML interface on a server running IIS version 7.0, you must enable **CGI modules** and **ISAPI modules**. Parent Paths must also be enabled, either for the Default Web Site as a whole, or for the RFXML virtual directory which appears after the installation of the Java/XML API.

Installing the Java Interface

Follow the instructions for installing the RightFax server in the *RightFax Installation Guide* and use the following specific steps:

- On the **Setup Type** screen, select **Custom** and then click **Next**.
- On the **Setup Features** screen, expand the **RightFax Server** heading in the components tree and select the **Java/XML API** component to install. Click **Next**.

Complete the remaining instructions in the *RightFax Installation Guide* for installing the RightFax server. After the RightFax server and Java/XML API component are installed and activated, you must configure an SMTP host and an IIS user account.

SMTP Host. Open the RightFax Server Module and click the e-transport tab. Enter the name of the SMTP server on your network that will transport all SMTP alerts and notifications regarding the RightFax server. If you will not be using SMTP to deliver RightFax alerts and notifications, you can leave this option blank.

IIS User Account. In IIS, configure the **rfxml** website with an IIS user account that RightFax will use to access the IIS server. This account is required for Java development.

Additional Java development tools are located in the Program Files\RightFax\Production\XML\Java folder.

Configuring the Java development computer to use the RightFax Java interface

Copy RFJavalnt.zip from the Program Files\RightFax\Production\XML\Java folder on the RightFax server to the development computer or the computer that will be running the Java interface. If you already have Java classes located on this computer, Captaris recommends copying this file to the same location. When the file has been copied, you must modify the CLASSPATH environment variable to include this .zip file.

Example If the RFJavalnt.zip file is located in the C:\Proj\Java folder on the development computer, then set the following CLASSPATH environment variable:

```
CLASSPATH=%CLASSPATH%;C:\Proj\Java\RFJavalnt.zip
```

Finally, when using the RightFax Java class library, you must import the RightFax class package by including this line in your Java code:

```
import RightFax.*;
```



Important *Because Java is case-sensitive, this line must be included in your Java code using the exact capitalization in this example. Incorrect capitalization will fail to import the RightFax Java class library.*

Sending Outbound Documents

To submit an outbound document, you must set sender information, recipient information, and add the content for the primary document (if any). These are encapsulated in an *RFDoc* object, in the *RFaxSubmit* object. Then you can add any attachments. The following list describes the basic steps.

1. Create an *RFaxSubmit* object.
2. Set the target URL for the RightFax server, using the *setTargetURL* method. This takes a *String*, NOT a URL object (it will create the URL object).

3. Use the *RFDoc* object, called *m_FaxDocument*, in the *RFaxSubmit* object to set all the non-attachment information about the document.
 - Call *m_FaxDocument.setSenderInfo (...)* to set the information about the person sending the document. This method has been overloaded, with a version that takes all information as parameters, a version that only takes the minimum information (*RFUser ID*), and versions that take the more common information. If there is not an overloaded version with exactly what you want, use one that takes more, and set the parameters you don't want to use to empty strings (*""*).
 - Call *m_FaxDocument.addRecipient (...)* for each recipient. Each time you call this, a recipient will be added to the list. This method has been overloaded, with a version that takes all information as parameters, a version that only takes the minimum information (*Fax Number*), and versions that take the more common information. If there is not an overloaded version with exactly what you want, use one that takes more, and set the parameters you don't want to use to empty strings (*""*).



Note *New to version RightFax 8.0, methods have been added to set e-mail and printer recipients. See the JavaDocs for details.*

- Call *m_FaxDocument.setBody (...)* to set the contents of the fax body (if any). If the body data is not plain text, make sure to set the type as well.
 - Call *m_FaxDocument.setCoverText (...)* to set the contents of the fax cover sheet text (if any). If the cover text is a data type other than plain text, then use *m_FaxDocument.setCoverTextType (...)* to set the type.
4. If you have any attachments, make a call to *addAttachment (...)* for each attachment. Each time you call this, an attachment is added to the list. This method takes a fully qualified path as its parameter, not the contents of the file to be attached.

- When all the data is set, call the method. You will get back a Vector of RFSStatus objects. Each object will contain the status for one recipient, stating whether it was passed on for sending or if there was an error.



Note You can call the `submitEx` method in place of `submit`. This will return a string containing the unparsed XML returned by RightFax. The XML contains the status of each recipient in the form of the following example (page 168).

Sample code for creating an outbound document

```
import RightFax*;
import java.net.MalformedURLException;
import java.net.UnknownHostException;
import java.io.IOException;
import java.util.*;

class FaxSubmit
{
    //Create a outbound fax object
    RFaxSubmit obFS = new RFaxSubmit();
    //Set the URL of the RightFax server
    obFS.setTargetURL ("http://www.company.com/");
    //Set the information on who is sending the fax
    obFS.m_FaxDocument.setSenderInfo ("Bill Smith", "", "Acme,Co.", "", "", "", "", "", "", "BILLS");
    //Add 2 recipients
    try {
        obFS.m_FaxDocument.addRecipient ("PRODXML:0001", "555-1234", "", "Jim Jackson", "", "", "", "", "", "");
    } catch (RFNoFaxNumberException nfne) {
        System.out.println (nfne.toString());
    } catch (RFInvalidIDException iide) {
        System.out.println (iide.toString());
    }
    //Add an e-mail and a printer recipient
    try {
        obFS.m_FaxDocument.addRecipient_email ("EMAIL:00000001", "smithj@company.com", "", "Here is the document", "");
        obFS.m_FaxDocument.addRecipient_printer ("PRNT:000000001", "MyPrinter", (short)1);
    } catch (RFNoDestinationException nde) {
```

```

        System.out.println (nde.toString());
    } catch (RFInvalidIDException iide) {
        System.out.println (iide.toString());
    }
    //Set the body text
    obFS.m_FaxDocument.setBody ( "Here is some body text", "TXT", -1, -1, -1, "Arial", -1, -1);
    //Set the cover text
    obFS.m_FaxDocument.setCoverText ("Here is some cover text");
    //Add attachments
    obFS.addAttachment ("c:\\documents\\mydoc.doc");
    obFS.addAttachment ("c:\\documents\\license.pdf");
    //Send the document, and get back the results.
    Vector obRetList = null;
    try {
        obRetList = obFS.submit();
    } catch (MalformedURLException mue) {
        System.out.println (mue.toString());
    } catch (UnknownHostException uhe) {
        System.out.println (uhe.toString());
    } catch (IOException ioe) {
        System.out.println (ioe.toString());
    } catch (RFNoDataException nde) {
        System.out.println ("Error:" + nde.toString());
    }
    //Output the results
    int nSize = obRetList.size();
    for (int i = 0; i < nSize; i++)
    {
        RFStatus obStat = (RFStatus)(obRetList.get(i));
        System.out.println ((i+1) + "-");
        System.out.println ("\tID: " + obStat.getID());
        System.out.println ("\tStatusCode: " + obStat.getStatusCode());
        System.out.println ("\tStatusMessage: " + obStat.getStatusMsg());
    }
}

```

Querying Documents

To perform a query on the status of a document (or group of documents), you must set the criteria for each query. The following list describes the basic steps.

1. Create an `RFaxQuery` object.
2. Set the target URL for the RightFax server, using the `setTargetURL` method. This takes a *String*, not a URL object (it will create the URL object).
3. Call `addQuery (...)` for each query you want to add to the request. This method has been overloaded, with a version that takes all information as parameters, and versions that take the more common information. If there is not an overloaded version with exactly what you want, use one that takes more, and set the parameters you don't want to use to empty strings ("") or *NULL* for *CALANDER* objects. You must set at least one query criteria parameter.
4. When all the queries are set, call the submit method. You will get back a Vector of `RFStatus` objects. Each object will contain the status of one document (not one query—a query could return many fax statuses).

Sample code for querying a document

```
import RightFax*;
import java.net.MalformedURLException;
import java.net.UnknownHostException;
import java.io.IOException;
import java.util.*;

class FaxQuery
{
    //Create a RFaxQuery object
    RFaxQuery obQ = new RFaxQuery();
    //Set the URL of the RightFax server
    obQ.setTargetURL ("http://www.company.com/");
    //Criteria for one query.
```

```

try {
    obQ.addQuery ("PRODXML:0001", null, null, "", "", "");
} catch (RFEmptyQueryException eqe) {
    System.out.println (eqe.toString());
} catch (RFInvalidIDException iide) {
    System.out.println (iide.toString());
}
//Send the query, and get back the results
Vector obQRetList = null;
try {
    obQRetList = obQ.submit();
} catch (MalformedURLException mue) {
    System.out.println (mue.toString());
} catch (UnknownHostException uhe) {
    System.out.println (uhe.toString());
} catch (IOException ioe) {
    System.out.println (ioe.toString());
} catch (RFNoDataException nde) {
    System.out.println (nde.toString());
}
//Output the results
int nQSize = obQRetList.size();
for (int i = 0; i < nQSize; i++)
{
    RFStatus obStat = (RFStatus)(obQRetList.get(i));
    System.out.println ((i+1) + "-");
    System.out.println ("\tID: " + obStat.getID());
    System.out.println ("\tStatusCode: " + obStat.getStatusCode());
    System.out.println ("\tStatusMessage: " + obStat.getStatusMsg());
}
}

```

Performing Actions on Documents

To perform an action on a document, you need the unique ID for that document. Once you have that, you can either delete the document, forward it to another recipient, or use it to create a library document. If you don't have the unique ID, you can obtain it by doing a query on the information that you do have (see [“Querying Documents”](#) on page 170).

When you submit a document, you have the option of creating a unique ID for that document or letting the RightFax server assign a unique ID for you.



Note *If you create your own unique ID, it must be 15 characters or less.*

For unique IDs that the software creates, the format is:

- The first seven characters are the name of the RightFax server.
 - The last eight characters are a number unique to a document.
1. Create an *RFaxAction* object.
 2. Set the target URL for the RightFax server, using *setTargetURL* method. This takes a *String*, NOT a URL object (it will create the URL object).
 3. Call *addForwardAction (...)* for each *forward* action you want to perform. This method has been overloaded, with a version that takes all information as parameters and a version that only takes the minimum information (ID and fax number). If there is not an overloaded version with exactly what you want, use one that takes more, and set the parameters you don't want to use to empty strings (“”).
 4. Call *addDeleteAction (...)* for each *delete* action you want to perform. The *ID* parameter is required.
 5. Call *addLibDocAction (...)* for each *create library document* action you want to perform. The *ID*, *LibDocID*, and *LibDocDescription* parameters are all required.
 6. When all the actions are set, call the submit method. You will get back a Vector of *RFStatus* objects. Each object will contain the status of one action, stating whether or not the action was successful.

Sample code for performing an action on a document

```
import RightFax.*;
import java.net.MalformedURLException;
import java.net.UnknownHostException;
import java.io.IOException;
import java.util.*;

class FaxAction
{
    //Create a RFaxAction object
    RFaxAction obA = new RFaxAction();
    //Set the URL of the RightFax server
    obA.setTargetURL ("http://www.company.com/");
    //Create a forward action
    try {
        if (!obA.addForwardAction("PRODXML:0001", "555-6789", "Mike Michell", "", "", "", "")) {
            System.out.println ("Add Forward Action Failed");
        }
        catch (RFNoFaxNumberException nfne) {
            System.out.println (nfne.toString());
        }
        catch (RFNoIDException nide) {
            System.out.println (nide.toString());
        }
        catch (RFInvalidOpException ioe) {
            System.out.println (ioe.toString());
        }
    }
    //Create a delete action
    try {
        if (!obA.addDeleteAction("PRODXML:0002")) {
            System.out.println ("Add Delete Action Failed");
        }
        catch (RFNoIDException nide) {
            System.out.println (nide.toString());
        }
        catch (RFInvalidOpException ioe) {
            System.out.println (ioe.toString());
        }
    }
    //Send the action requests, and get back the results
    Vector obARetList = null;
}
```

```

try {
    obARetList = obA.submit();
} catch (MalformedURLException mue) {
    System.out.println (mue.toString());
} catch (UnknownHostException uhe) {
    System.out.println (uhe.toString());
} catch (IOException ioe) {
    System.out.println (ioe.toString());
} catch (RFNoDataException nde) {
    System.out.println (nde.toString());
}
}
//Output the results
int nASize = obARetList.size();
for (int i = 0; i < nASize; i++)
{
    RFStatus obStat = (RFStatus)(obARetList.get(i));
    System.out.println ((i+1) + "-");
    System.out.println ("\tID: " + obStat.getID());
    System.out.println ("\tStatusCode: " + obStat.getStatusCode());
    System.out.println ("\tStatusMessage: " + obStat.getStatusMsg());
}
}

```

Using Debug Mode

The method *setDebug* is part of the *RFax* class and is therefore callable from any *RFaxSubmit*, *RFaxAction*, or *RFaxQuery* object.

By turning on debug mode, you can get more information sent to the standard out. This includes information such as the XML generated for sending to the RightFax server and the XML returned from the RightFax server.

To turn on debug mode, call the *setDebug* method, passing it *true*.

Error and Status Codes for XML- and Java-Based Documents

The RightFax API for Java will return status information (as a vector of RFStatus objects or as a string containing XML) for all operations, unless an error occurs in the API, in which case an exception will be thrown.

Each RFStatus object or XML node will contain a status code, status message, and document ID.

The following table lists all possible codes, their associated messages, and an explanation for what each indicates.

Table 25b XML Interface Error and Return Codes

Code	Message	Explanation
General		
-1	Failed to load XML into DOM tree. XML file copied to err dir.	The XML file could not be loaded into the DOM tree. This is usually caused by the file not being found, a syntax error in the XML file, or the XML file not conforming to the schema.
-3	Failed to connect to RF Server.	The production software was unable to connect to the RightFax server. This is usually caused by the RFServer service not running.
-5	Unknown XML operation type.	The operation was not XML_FAX_SUBMIT, XML_FAX_QUERY, or XML_FAX_ACTION.

Table 25b XML Interface Error and Return Codes (Continued)

Code	Message	Explanation
Submit		
-2	Failed to load XSL into DOM tree.	The XSL file could not be loaded into the DOM tree. This is usually caused by the file not being found or syntax error in the XSL file.
-4	Failed to get XSL file info.	The XML file did not contain information on the XSL file to use, and the default (in the registry) could not be found.
0	Submit failed.	Unable to perform the submit requested. The message will contain the reason.
1	Document has been successfully submitted for sending.	The document has been passed on for sending. If notifications are set up, you will be informed of the success or failure of the send via that notification type.
Query		
-99	No information found.	The query returned no documents matching your criteria.
0	Query failed.	Unable to perform the query requested. The message will contain the reason.

Table 25b XML Interface Error and Return Codes (Continued)

Code	Message	Explanation
0-n	N/A	If you receive a message not listed in this table, it is a code universal to all RightFax software. These are listed in the <i>RightFax Administrator's Guide</i> .
6	Deleted	If the document had previously been sent successfully, but has since been deleted.

Action

-99	No such fax found.	The document you wish to perform the action on could not be found.
0	Action failed.	Unable to perform the action requested. The message will contain the reason.
1	Fax (<ID>) has been deleted.	Success
1	Fax (<ID>) has been submitted for forwarding to: <recipient>.	Success
1	A Library Document has been created from fax: <ID>.	Success

Notification

0-n	N/A	If you receive a message that is not listed in this table, it is listed in the <i>RightFax Administrator's Guide</i> .
-----	-----	--

■ ■ ■

Creating FCL Documents with InternetLink Commands

The RightFax InternetLink Module is a component of the RightFax Enterprise Integration Module. The InternetLink Module enables the Integration Module to build documents in Multipurpose Internet Mail Extensions (MIME) or text format and send them via Simple Mail Transfer Protocol (SMTP) through the Internet.

You can use the InternetLink Module to send documents in native mode or filter mode.

System Requirements

The InternetLink Module requires the following:

- RightFax Enterprise Server and Integration Module already installed.
- Network connection to the Internet.
- SMTP gateway on the network.
- TCP/IP connection from the RightFax Integration Module to the network

Activating the InternetLink Module

The files required by the InternetLink Module are installed on all RightFax servers during the server installation. However, the InternetLink Module must be licensed and activated before its functionality will be enabled.

To activate the InternetLink Module, you must have licensed a RightFax server type that includes this connector, or purchased and licensed this connector separately. For information on activating new components on the RightFax server, refer to the *RightFax Installation Guide*.

Before you begin sending documents with the InternetLink Module, verify the name of your e-mail server.

To verify the name of your e-mail server

1. On the RightFax server, select **Start > Programs > Enterprise Fax Manager**. The Enterprise Fax Manager window opens.
2. Select the RightFax server in the list of servers in the left pane of the window. A list of services appears in the lower-right pane.
3. In the **Service Name** list, double-click **RightFax Server Module**. The **Server Configuration** dialog box opens.
4. Click the **eTransport** tab.

- Verify that the fully qualified domain name of your SMTP server appears in the **SMTP Hostname** box. If the name of your server is not correct, enter the correct name.

To create and send documents with the InternetLink Module, use the InternetLink FCL commands.

Understanding the InternetLink FCL Commands

Documents sent via the InternetLink Module require six FCL commands. Without all of these commands, the document will not be sent. The required commands are listed in the following table.

Table 26a Required FCL Commands

Command	Description
{{begin}}	Indicates the beginning of a document. The Integration Module will process all the data that appears between a {{begin}} and {{end}} command as a discreet document. Data that does not appear between the {{begin}} and {{end}} commands is ignored. This command must appear as the first command in each InternetLink document.
{{end}}	Indicates the end of a document. The Integration Module will process all the data that appears between a {{begin}} and {{end}} command as a discreet document. Data that does not appear between the {{begin}} and {{end}} commands is ignored. This command must appear as the last command in each InternetLink document.
{{from}}	Sender's e-mail address.
{{subject}}	Subject line of the e-mail message.

Table 26a Required FCL Commands (Continued)

Command	Description
{{to}}	Recipient's e-mail address.
{{type email}} or {{type mime}}	{{type email}} sends a document as the editable body of an e-mail message. {{type mime}} sends a document as an attachment to an empty e-mail message.

The commands {{begin}} and {{type}} can be replaced with a "shortcut" command, {{begin mime}} or {{begin email}}.

Choosing document types

The document type determines the type of document that the recipient receives via e-mail. The type of document is specified with the {{type}} command.

- {{type email}} converts the document to the editable body text of an e-mail message. This is plain text; it has no formatting.
- {{type mime}} converts the document to an un-editable, MIME-encoded graphic attachment to an empty e-mail message. This retains all formatting and graphics. Sending documents in MIME format is best when the document must not be editable or when it must be an exact replica of a pre-printed, hardcopy form.

Choosing image types

The image type refers to the type of graphic file that is created when you use the {{type mime}} command, PDF, TIF, or PCX. To specify an image type, use the {{imagetype}} command (see "IMAGETYPE" on [page 185](#)).

The default graphic format for {{type mime}} documents is Group 4 TIFF.

Using include files with the InternetLink Module

When you activate the InternetLink Module, an empty include file called Mime.inc is created in the RightFax\Production\Include folder. You can insert plain text (but not FCL, formatted text, or graphics) in Mime.inc, and this text will become the body of a `{{type mime}}` document.

Mime.inc is linked by default to all InternetLink documents. Because it is empty, Mime.inc has no function unless you add information to it. If you add information to Mime.inc, then that information will appear in the body of each `{{type mime}}` message that is sent using the InternetLink Module.

For information on other types of include files, see the *RightFax Integration Module Guide*.

Attaching Native Files to InternetLink Documents

Without the InternetLink Module, the Integration Module converts all attachments to fax format (a TIF image) before sending. The InternetLink Module gives you the option to attach documents in native file format. For example, a Microsoft Word document can be sent as a Word document.

To attach a file, use the `{{attach}}` or `{{beginnative}}` commands. For a description of both commands, see, “[Standard FCL Commands with InternetLink Options](#)”.

In the following example, Program.xls will be attached in its native format (as a Microsoft Excel document) to the host document.

Example `{{attach c:\\IST Files\\Programs.xls native}}`

If you create a `{{type mime}}` document that contains the `{{attach}}` command with the native option, then the attached document becomes the second attachment to an empty e-mail message.

If you use the `{{attach}}` command without the native option, then the attached document and the host document are merged into one graphic file that is attached to an empty e-mail message. You determine the image type of this graphic file with the `{{imagetype}}` command “[Choosing image types](#)” on [page 178](#).

Use the native option to the `{{attach}}` command when the file must be editable or when fax format cannot adequately represent the content of a file. For example, fax format cannot adequately represent the content of an audio file.

You can attach multiple documents by inserting multiple `{{attach}}` commands. The `{{attach}}` command cannot be used with the `{{type email}}` command.

Example FCL for InternetLink documents that have attachments

The following examples illustrate FCL documents with attachments created for the InternetLink Module. In each example, the FCL (on the left) yields different final results (on the right).

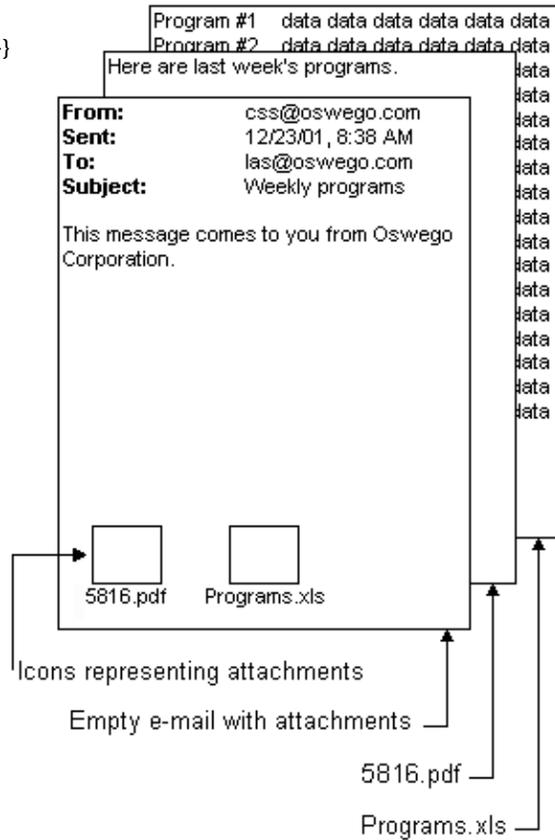
In the following example, the host data (“Here are last week’s programs.”) is converted to PDF and is saved as the attachment called 5816.pdf (the file name is generated automatically by the system) through the use of the `{{type mime}}` and `{{imagetype pdf}}` commands. The attached document (Programs.xls) is also attached in its native format (a Microsoft Excel file) through the use of the `{{attach}}` command with the native option.

The example includes text that was added to Mime.inc, “This message comes to you from Oswego Corporation”. For more information, “[Using include files with the InternetLink Module](#)” on [page 179](#).

Figure 26.1 Attaching a File in its Native Format

```

{{begin}}
{{type mime}}
{{imagetype pdf}}
Here are last week's programs.
{{attach "c:\IST Files\Programs.xls" native}}
{{to css@oswego.com}}
{{subject Weekly programs}}
{{from las@oswego.com}}
    
```



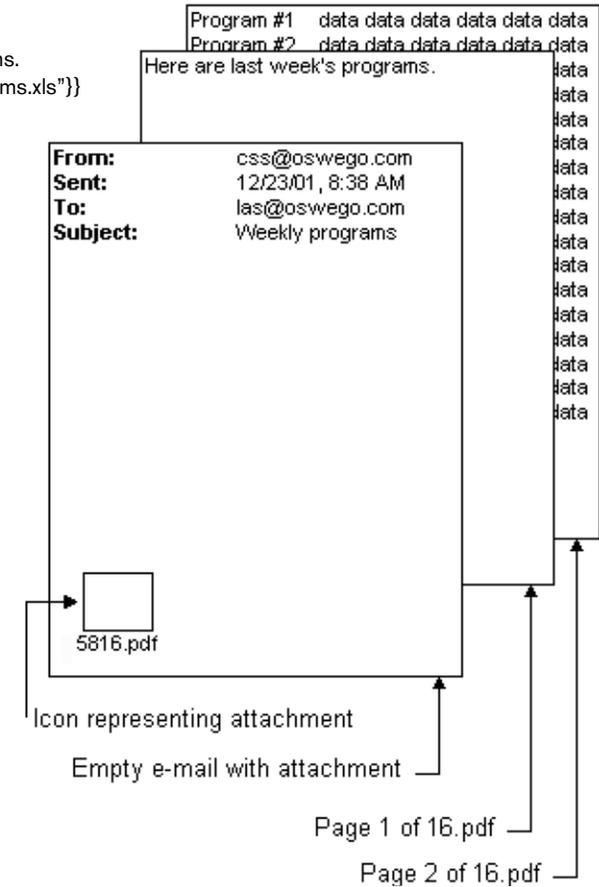
In the following example, the host data (“Here are last week’s programs.”) is converted to PDF and becomes page 1 of the attachment called 5816.pdf through the use of the `{{type mime}}` and `{{imagetype pdf}}` commands.

Because the native option of the `{{attach}}` command was not used, Programs.xls is converted to PDF and becomes page 2 of 16.pdf.

Figure 26.2 Attaching a File Converted to PDF Format

```

{{begin}}
{{type mime}}
{{imagetype pdf}}
Here are last week's programs.
{{attach "c:\IST Files\Programs.xls"}}
{{to css@oswego.com}}
{{subject Weekly programs}}
{{from las@oswego.com}}
{{end}}
    
```



Sending Documents as E-mail if Faxes Fail

With the InternetLink Module, you can send e-mail documents if fax numbers fail. To do this, use the `{{onerror}}` command.

The `{{onerror}}` usage options relevant to the InternetLink Module are:

```
{{onerror {email|mime e-mail address}}}
```

Unlike the `{{type}}` command, the e-mail and mime options for the `{{onerror}}` command do the same thing. With either option, the document is sent as a graphic image attached to an empty e-mail message. With both options, you must use the `{{imagetype}}` command to specify the type of graphic image to create.

In the following example, if the document fails to send as a fax, then it will be sent as a PDF file attached to an empty e-mail message to `css@oswego.com`.

```
{{begin}}  
{{fax 503-555-4489}}  
{{onerror email css@oswego.com}}  
{{imagetype pdf}}  
Body of the document to be sent.  
{{end}}
```

When a fax fails and is sent as an e-mail, all e-mail addressing options (such as `{{to}}` and `{{cc}}`) in the original document are replaced with the e-mail address specified in the `{{onerror}}` command. The content of the `{{contact}}` command is replaced with "To whom it may concern." All other fields, such as `{{subject}}` or `{{from}}`, are retained as they were in the original document.

Receiving Notification When a Fax Fails and Is Sent as an E-mail

To create a notification that a fax has failed and has been sent as an e-mail, include the `^type^` keyword in the notification template. For more information on notification messages, refer to ["Setting Up Notification Messages of Document Transmission"](#)

Standard FCL Commands with InternetLink Options

The following standard FCL commands have options that can be used with the InternetLink Module. The commands are described in detail in the *RightFax Integration Module Guide*

ATTACH

The `{{attach}}` command ends the current page (unless it is blank) and attaches one or more files, optionally deleting them after they have been added to the current document.

This command can specify that an attached file be sent in its native format, such as Word or Excel). Include the "native" option in the command, as illustrated in the following example.

The `{{attach}}` command cannot be used with the `{{type email}}` command.

Example`{{attach "C:\Pricing\PriceList.xls" native}}`

ONERROR

The `{{onerror}}` command specifies an action for the document in the event that it fails in transmission. For example, the document can be faxed or e-mailed to alternate recipients.

ONSUCCESS

The `{{onsuccess}}` command specifies an action for the document in the event that it is successfully transmitted. For example, a copy of the document can be faxed or e-mailed to alternate recipients, such as the original sender of the document.

TYPE

The `{{type}}` command establishes the type of document that should be sent.

- The `{{type email}}` command specifies that the document will be sent as the editable body text of an e-mail message.
- The `{{type mime}}` command specifies that the document will be sent as an un-editable, MIME-encoded graphic attachment to an empty e-mail message.

If you choose `{{type mime}}`, then you must also use the `{{imagetype}}` command to specify the type of MIME-encoded graphic attachment to create (PCX, TIF, or PDF).

If you use the `{{begin mime}}` or `{{begin email}}` commands, do not use the `{{type}}` command.

InternetLink FCL Commands

The following commands are specific to the InternetLink Module.

BEGIN MIME or BEGIN EMAIL

Syntax `{{begin {mime|email}}}`

Examples `{{begin email}}`
`{{begin mime}}`

The `{{begin mime}}` and `{{begin email}}` commands specify the transmission type. They can replace the `{{begin}}` and `{{type}}` commands.

BEGINNATIVE

`{{Beginnative}}` is similar to the `{{attach}}` command with the native option. Both commands let you send documents in native format. The `{{beginnative}}` command also lets you specify advanced configuration options that `{{attach}}` does not support. The key difference between `{{beginnative}}` and `{{attach}}` is that `{{beginnative}}` requires you to put the content of the to-be-attached file in the FCL itself, while `{{attach}}` attaches a file from a different location.

The body option of `{{beginnative}}` inserts the data (which appears between `{{beginnative}}` and `{{endnative}}` commands in your host data stream) into the body of an e-mail message.

The inline option suggests to the recipient e-mail client that the data between `{{beginnative}}` and `{{endnative}}` commands be displayed as an embedded object in the body of the e-mail message. If the recipient e-mail client cannot display the host data in this way, then the data will be sent as an attachment that is represented by an icon in the body of the e-mail message. The icon can be opened with the appropriate application.

`{{Beginnative}}` requires `{{endnative}}`.

Syntax `{{beginnative "filename" [body] [inline] [mediatype=type] [base64|quotedprintable]}}`

Example `{{beginnative "Body.txt" body mediatype=text/plain}}`

The example creates an attachment called Body.txt (whose content is the host data between the `{{beginnative}}` and `{{endnative}}` commands in the FCL document). The attachment will be displayed as editable text in the body of an e-mail message.

Configurable attributes of `{{beginnative}}`:

- File name, described on [page 183](#)
- Body, described on [page 183](#)
- Inline, described on [page 183](#)
- Mediatype, described on [page 184](#)
- Base64 or quotedprintable, described on [page 184](#)

File name

This is the suggested file name for the data. You need not supply a path with this file name because the file data is stored between the `{{beginnative}}` and `{{endnative}}` commands. This is the only required parameter of `{{beginnative}}`.

Example `{{beginnative "File.pdf" inline
mediatype=application/pdf}}
Here are the files you wanted
{{endnative}}`

In this example, RightFax will suggest to the recipient e-mail client that "Here are the files you wanted" be displayed as an embedded object in the body of the e-mail message. If the recipient e-mail client cannot perform this task, then an icon labeled File.pdf will appear in the e-mail message. The content of File.pdf will be "Here are the files you wanted."

Body

This sends the data that appears between the `{{beginnative}}` and `{{endnative}}` commands as text in the body of the e-mail message.

Using the body option, you can specify alternative file formats when you want to choose the format that the recipient's e-mail software should display.

For example, you can send a text file and an HTML file. Generally, the recipient's e-mail software will display the file in the format that it supports. It will not display both files.

The sequence of the file formats determines the format that the recipient's e-mail software will display. Generally, it will display the first file format that it supports. For example, if you specify text and HTML format, a software program that supports both formats will display text first.

To specify alternative formats, use multiple `{{beginnative}}` commands with the body option.

Example To send a document that contains both HTML and text, you might use the following FCL:

```
{{begin mime}}  
{{to drew@MetroCleaners.com}}  
{{from sarah@MetroCleaners.com}}  
{{subject Sample of text and HTML in the body of the  
message}}  
{{beginnative "body.txt" body mediatype=text/plain}}  
Drew,  
Here are the files that you asked for.  
{{endnative}}  
{{beginnative "body.htm" body mediatype=text/html}}  
<HTML><HEAD><TITLE>/TITLE><META  
content="text/html; charset=iso-8859-1"  
http-equiv=Content-Type>/HEAD>  
<BODY><HR><STRONG>Drew,</STRONG><BR>  
Here are the file that you asked for.<HR>  
</BODY></HTML>  
{{endnative}}  
{{end}}
```

Inline

This suggests to the recipient e-mail client that the data between `{{beginnative}}` and `{{endnative}}` commands be displayed as an embedded object in the body of the e-mail message. If the recipient e-mail client cannot display the host data in this way, then the data will be sent as an attachment that is represented by an icon in the body of the e-mail message. The icon can be opened with the appropriate application.

Mediatype

Specifies the nature of the data between `{{beginnative}}` and `{{endnative}}` commands. The content type (which is required if you use the `mediatype` option) is represented by a top-level mediatype, a slash, and a subtype.

Example `text/plain`

The top-level media type identifies a general type of data, and the subtype identifies a specific format for that type of data. Thus, a media type of `image/tif` is enough to tell a recipient e-mail client that the data is an image, even if the recipient e-mail client cannot interpret the specific image format of `.tif`.

The default mediatype for native documents with the `body` option is `text/plain`. Without the `body` option, the default mediatype is `application/octet-stream`. Other common mediatypes are (but are not limited to):

- `application/ms-word`
- `application/pdf`
- `audio/mid`
- `audio/wav`
- `image/gif`
- `text/html`
- `text/xml`
- `video/mpeg`

If you do not supply a media type, or if you supply a media type that the recipient e-mail client cannot interpret, the document most likely will not fail because most current e-mail clients can detect the media type.

Base64 or Quotedprintable

The data between `{{beginnative}}` and `{{endnative}}` commands will be encoded using the specified scheme. This does not necessarily require that the attachment be encoded using this scheme in the e-mail that is sent to the recipient. If these options are not specified, then the data will not be encoded (8 bit).

CC

Stores the SMTP address for a carbon copy recipient of a document. To store multiple addresses, repeat this command for each recipient.

Syntax `{{cc address}}`

Example `{{cc Justin@OKtires.com}}`
`{{cc Tommy@OKtires.com}}`

The examples send a copy of the e-mail message to both `Justin@OKtires.com` and `Tommy@OKtires.com`.

ENDNATIVE

Ends the processing of an attached native document that was begun with the `{{beginnative}}` command.

Syntax `{{endnative}}`

Example `{{endnative}}`

FROM

Stores the return SMTP address of the sender of the document.

Syntax `{{from address}}`

Example `{{from Donny@Pacific.com}}`

The example uses `Donny@Pacific.com` as the return e-mail address for the e-mail message.

IMAGETYPE

Selects the graphic format of a `{{type mime}}` document.

Syntax `{{imagetype {pdf|group3|group4|pcx}}}`

Example `{{imagetype pdf}}`

The example identifies PDF as the graphic format for a `{{type mime}}` document.

SUBJECT

Stores the subject of an SMTP document. The subject appears in the subject field of the e-mail message.

Syntax `{{subject topic}}`

Example `{{subject RightFax InternetLink example.}}`

The example creates an e-mail message with a subject field that says, "RightFax InternetLink example."

TO

Stores the SMTP address for the recipient of an SMTP document.

Syntax `{{to address}}`

Example `{{to John@OregonManufacturers.com}}`

The example sends the e-mail message to John@OregonManufacturers.com.

■ ■ ■

Appendix A

Integration Module Programs

The following table lists the programs that are installed with the RightFax Integration Module in the RightFax\Production\Bin folder. You can use them to help you configure and fax-enable your system. To see a list of the command line options for each program, type **-h** at the command prompt to view online help.

Table A1 Programs Installed With the Integration Module

Program	Description	Command line syntax
Bufile.exe	Reads a directory and executes a command on each file that is placed in the directory.	bufile [options] <i>directory</i>
Capture.exe	Reads input from a serial port at different rates and in different modes and executes a command on data that is received.	capture [options] <i>input</i>
Dbnotify.exe	Updates databases from a Win32 command line.	dbnotify [options] <i>Data Source Name</i>
Diagtiff.exe	Diagnoses TIFF image files and provides the contents of their tags.	diagtiff [options] <i>infile [infile ...]</i>
Encode64.exe	Encodes files using base64 encoding.	encode64 [options] <i>infile outfile</i>
Fax2mapi.exe	Sends MAPI32 messages from the Win32 command line.	fax2mapi [options]

Table A1 Programs Installed With the Integration Module (Continued)

Program	Description	Command line syntax
Fax2note.exe	Sends messages to Lotus Notes.	fax2note [options]
Hd.exe	Displays files in hexadecimal format.	hd [options] <i>input file</i>
Hf.exe	Sends data to a TCP/IP port.	hf [options]
Hlpisend.exe	Sends data via HLLAPI. Requires emulation software that supports HLLAPI. Capable of using a log-on script to log in to a 3270 session prior to transmitting data.	hlpisend [options]
Makedoc.exe	Submits an FCL-encoded file to the Buffer.exe program. Buffer.exe reads the file, parses it for syntax, notes any errors, and passes the file to Parse.exe. Parse.exe prepares the document to be sent.	makedoc [options] <i>filename</i>
Multitiff.exe	Breaks one multiple-page TIFF image into a document consisting of single-page TIFF images.	multitiff [options] <i>documentnumber startingpage</i>
Nplisten.exe	Creates a named pipe and executes a command on data received via the named pipe.	nplisten [options]

Table A1 Programs Installed With the Integration Module (Continued)

Program	Description	Command line syntax
Portlstn.exe	Executes a command on a data stream received via a TCP/IP port.	portlstn [<i>options</i>] <i>command</i>
Pssplit.exe	Splits a PostScript file to create separate pages.	pssplit [<i>options</i>] <i>infile</i> <i>outputdirectory</i>
Tee.exe	Transcribes standard input to standard output and makes copies in the file.	tee [<i>options</i>] <i>outputfile</i>
Tiffbind.exe	Combines single-page TIFF images into one multiple-page TIFF image.	tiffbind [<i>options</i>] <i>outputtiff inputtiff {input tiff ...}</i>
Tog3.exe	Converts differing file formats of black and white TIFF images to Group 3, Group 4, PCX, and GIF images.	tog3 [<i>options</i>] <i>infile</i> <i>outfile</i>
Uuencode	Encodes files using uuencode.	uuencode [<i>options</i>] <i>infile outfile</i>

■ ■ ■

Appendix B

FCL Commands

This appendix provides a list of the facsimile command language (FCL) commands used to create FCL documents. This list provides detailed information on the command syntax and usage.

ABORT

Causes the software to ignore a document before it is faxed. You can place this command anywhere in the FCL document.

Syntax `{{abort}}`

Example `{{abort}}`

ADDCOPIES

Increments the number of copies to print. This affects documents with the `{{type print}}` command.

Syntax `{{addcopies number}}`

Example `{{addcopies 3}}`

The example increments the number of printed copies of the current document by three.

See also “TYPE” on [page 216](#)

ALTFAX

Stores an alternate fax number for document if the primary fax number fails.

The `{{altfax}}` argument is cleared when a document is broadcast or the `{{execute}}` command is used. To broadcast each document with an alternate fax number, add an `{{altfax}}` command after every new `{{fax}}` or `{{execute}}` command.

Syntax `{{altfax fax number}}`

Example `{{altfax 503-555-5458}}`

The example sets the alternate fax number for the document to (503) 555-5458.

For the procedure that uses this command, see [Chapter 11](#), “Specifying the Destination of the Document”.

APPROVAL

Holds the document for approval in a FaxUtil mailbox.

Syntax `{{approval}}`

Example `{{approval}}`

For the procedure that uses this command, see [Chapter 20](#), “Scheduling Document Transmission”.

ATTACH

Ends the current page (unless it is blank) and attaches one or more files, optionally deleting them after they have been added to the document.

You can specify the full path to the file with the file name. The full path must be surrounded by quotation marks. The default is RightFax\Production\Forms.

You can use a wildcard with the file name. If the file name ends in * (such as sales.*), files matching the name and ending with any extension would be attached to the current document. This is useful for creating multiple page attachments.

If a document contains a `{{quality}}` command, it should be placed in the document before the `{{attach}}` command. The `{{attach}}` command cannot be used in `{{type email}}` documents.

The command can delete the attached file(s) after the fax is generated.

Syntax `{{attach filename [delete] [native]}}`

Examples `{{attach Polnv.doc delete}}`
`{{attach "C:\Program Files\RightFax\Production\MyDocs.doc" native}}`

The first example converts Polnv.doc to fax format (.tif) and attaches it to the document. After the fax is generated, Polnv.doc is deleted. Because no path is specified, Polnv.doc is located in the folder RightFax\Production\Forms.

The second example can be used with the InternetLink Module to send the document via e-mail. The base document would be sent as an e-mail, and Polnv.doc would be attached to that e-mail as a native file—it would be left as a Microsoft Word (.doc) file and not converted to fax format.

For the procedure that uses this command, see [Chapter 19](#), “Attaching and Embedding Files, Signatures, and Graphics”.

BATCH

Alters the scheduled time to the next round number of minutes. This is useful for keeping a document to send with a batch of other documents. The `{{batch}}` command differs from `{{delay}}` in that it rounds to the nearest number of minutes, rather than adding a specified number of minutes to the scheduled time. Specify the number of minutes to round to.

Syntax `{{batch minutes}}`

Example `{{batch 5}}`

The example batches to the next nearest five minutes. Therefore, if it is now 1:02 P.M., the batch time would be 1:05 P.M.

See also “DATE” on [page 195](#)
 “DELAY” on [page 195](#)
 “TIME” on [page 215](#)

For the procedure that uses this command, see [Chapter 20](#), “Scheduling Document Transmission”.

BEGIN

Starts an FCL document. Any text between this command and an `{{end}}` command is rendered as a single page for faxing or printing.

This command is required in every FCL document. FCL documents may contain multiple `{{begin}}` and `{{end}}` commands, but the FCL between each set of commands will be rendered as a separate page.

Syntax `{{begin}}`

Example `{{begin}}`

See also “END” on [page 196](#)

For the procedure that uses this command, see [Chapter 8](#), “Creating FCL Documents”.

BEGINCVT

This command embeds a document in an FCL document. The fax server converts the document to a TIF image using server side application (SSA) conversion. For more information on SSA, see the *RightFax Administrator's Guide*.

Alternatively, you can encode the embedded document using BASE64 or QUOTEDPRINTABLE. This command must be used with the command `{{endcvt}}`.

BASE64 is preferable to QUOTEDPRINTABLE unless the encoded data must be human-readable. If no encoding sequence is specified, the data is assumed to be 8-bit binary.

Syntax `{{begincvt FileName [base64|QuotedPrintable]}}`

Examples `{{begincvt proposal.doc}}`
`{{begincvt proposal.doc base64}}`

The first example establishes that binary data (in this case, a document called Proposal.doc) is included in the FCL data stream and must be converted.

The second example does the same as the first example but specifies to encode the data using BASE64 encoding.

Along with the required `{{endcvt}}` command, the FCL might look like this:

```
{{begincvt proposal.doc base64}} nnnnnnnn {{endcvt}}
```

In this example, nnnnnnnn is the actual binary data—the content of Proposal.doc—embedded in the FCL data stream. This data would be unreadable and undoubtedly much longer than the example here.

See also [“ENDCVT” on page 196](#)

BILLING

Specifies the billing code of the fax owner.

Syntax `{{billing code}}`

Example `{{billing 1234}}`

The example specifies that the billing code associated with this fax is 1234.

See also [“BILLING2” on page 191](#)

`{{Billing}}` translates to the RightFax server field billinfo1. For more information, [“Using Cover Sheets in a Broadcast” on page 69](#).

For the procedure that uses this command, see [Chapter 21](#), [“Setting Up Notification Messages of Document Transmission”](#).

BILLING2

Specifies a secondary billing code of the fax owner.

Syntax `{{billing2 code}}`

Example `{{billing2 4567}}`

The example specifies that the billing code associated with this fax is 4567.

See also [“BILLING” on page 191](#)

`{{billing2}}` translates to the RightFax server field billinfo2. For more information, [“Using Cover Sheets in a Broadcast” on page 69](#).

For the procedure that uses this command, see [Chapter 21](#), [“Setting Up Notification Messages of Document Transmission”](#).

BM

Sets the bottom margin for the current and subsequent pages. The size of the margin is specified in the current units from the bottom of the page.

Syntax `{{bm margin}}`

Example `{{bm 0.5}}`

The example sets the bottom margin to 1/2 a unit from the bottom of the page.

See also [“LM” on page 201](#)
[“TM” on page 215](#)
[“UNITS” on page 217](#)

For the procedure that uses this command, see [Chapter 16](#), [“Setting Margins and Tabs”](#).

BOX

Draws a box in the current line width at the specified coordinates, optionally filling it with text. The coordinates can be formatted (x,y) or (x y). The coordinates can be specified values, or you can specify a position relative to the current cursor position.

If only one coordinate pair is specified, the other coordinate pair is assumed to be the current position. Therefore, `{{box (5,5)}}` is equivalent to `{{box (x,y) (5,5)}}`.

If both coordinates are specified, you can specify a line of text to appear in the box. The text must be surrounded by quotation marks and is limited to one line. The text is placed in the box according to the current `placexy` settings, which default to the upper-left corner of the document.

Syntax `{{box coord1 coord2 [“text”]}}`

Examples `{{box (0,y) (7,7) “Hi There”}}`
`{{box 5 5 7 7}}`

The first example draws a box from the left side of the page at the current line to 7 units over, and 7 units down the current page, placing the text “Hi There” within it.

The second example draws a 2 by 2 unit box at the coordinates (5,5) to (7,7) in current units.

See also [“FILLBOX” on page 197](#)
[“LINEWIDTH” on page 201](#)
[“PLACEXY” on page 208](#)
[“RBOX” on page 210](#)
[“RFILLBOX” on page 211](#)
[“UNITS” on page 217](#)

For the procedure that uses this command, see [Chapter 15](#), [“Creating Lines, Boxes, and Other Shapes”](#).

CLEARTABS

Removes all tab stops from the document.

Syntax `{{cleartabs}}`

Example `{{cleartabs}}`

See also [“SETTAB” on page 213](#)
[“TAB” on page 214](#)

For the procedure that uses this command, see [Chapter 16](#), [“Setting Margins and Tabs”](#).

COMMENT

Stores a user-defined message specific to the document.

Syntax `{{comment comments}}`

Example `{{comment Inv# 12345}}`

The example associates the comment “Inv# 12345” with the document.

`{{Comment}}` translates to the RightFax server field `to_citystate`. For more information, [“Using Cover Sheets in a Broadcast” on page 69](#).

For the procedure that uses this command, see [Chapter 14](#), [“Creating and Attaching Cover Sheets”](#).

COMPANY

Stores a company name for the document.

Syntax `{{company name}}`

Example `{{company ABC Incorporated}}`

The example associates the company name “ABC Incorporated” with the document.

`{{Company}}` translates to the RightFax server field `to_company`. For more information, “[Using Cover Sheets in a Broadcast](#)” on [page 69](#).

For the procedure that uses this command, see [Chapter 14](#), “[Creating and Attaching Cover Sheets](#)”.

CONTACT

Stores the contact name for the document.

Syntax `{{contact name}}`

Example `{{contact Kim Boston}}`

The example associates the contact name “Kim Boston” with the document.

`{{Contact}}` translates to the RightFax server field `to_name`. For more information, “[Using Cover Sheets in a Broadcast](#)” on [page 69](#).

For the procedure that uses this command, see [Chapter 14](#), “[Creating and Attaching Cover Sheets](#)”.

COVER

Specifies a cover sheet template for the current document. You can specify the file name of a cover sheet template, or you can specify that a cover sheet should not be included.

The file name can be either a full path or a path relative to `RightFax\Production\Covers` (this path is for Integration Module cover sheets; RightFax server cover sheets are stored in a different

folder on the RightFax server). If the complete file name is not found, the RightFax Integration Module tries to open the file by adding a `.cov` extension. If this fails, the cover sheet is assumed to be a RightFax `.pcl` or `.doc` cover sheet. If no cover command is specified in a document, the default cover sheet defined in the RightFax Integration Module Configuration program is used.

Syntax `{{cover template}}`

Example `{{cover sales}}`

The example specifies the `sales` or `Sales.cov` file in `RightFax\Production\Covers` should be used as a template for the cover sheet of the current document.

Specific cover names

The `{{cover}}` command can contain the following special cover names:

- **None.** No cover sheet is generated, even if you have defined a default cover sheet in the RightFax Integration Module Configuration program.
- **Rfdefault.** The default cover sheet action associated with the user is performed.

Examples If a cover sheet is *not* defined for the user sending the fax:

`{{COVER NONE}}` – No cover sheet is sent.
`{{COVER RFDEFAULT}}` – No cover sheet is sent.
`{{COVER FCS.PCL}}` – Specified RightFax server (.pcl) cover sheet is sent.
`{{COVER PROD.COV}}` – Specified Integration Module (.cov) cover sheet is sent.

If a cover sheet *is* defined for the user sending the fax:

`{{COVER NONE}}` – No cover sheet is sent.
`{{COVER RFDEFAULT}}` – Default cover sheet is sent.
`{{COVER FCS.PCL}}` – Specified RightFax server (.pcl) cover sheet is sent.
`{{COVER PROD.COV}}` – Specified Integration Module (.cov) cover sheet is sent.

For the procedure that uses this command, see [Chapter 14](#), “Creating and Attaching Cover Sheets”.

COVERTEXT

This command creates text that should appear on the cover sheet. It stores subsequent input (until an `{{endcovertext}}` command ([page 196](#))) for later inclusion as a text block in the document cover sheet.

The Integration Module extracts the text between `{{covertext}}` and `{{endcovertext}}` commands. If PostScript commands appear in the text, they will not be stripped out. Therefore, do not embed PostScript between `{{covertext}}` and `{{endcovertext}}`.

`{{Covertext}}` translates to the RightFax server field `notetext`. For more information, “Using Cover Sheets in a Broadcast” on [page 69](#).

An index number is used to create a block of text within a single cover sheet. Each instance of a `^covertext` keyword in the cover sheet template will be replaced by the corresponding text enclosed between `{{covertext}}` and `{{endcovertext}}` commands. If no index number is specified, the text will be linked to the `^covertext 0` keyword.

Each instance of `{{covertext}}` must use a different index number (0 in the example), corresponding to a numbered `^covertext` field. 0 is the most frequently used number.



Note 0 (zero) is the only option for RightFax server (.pcl or .doc) cover sheets. For more information, “Using Cover Sheets in a Broadcast” on [page 69](#).

Syntax `{{covertext indexnumber}}`

Example `{{covertext 0}}`

For the procedure that uses this command, see [Chapter 14](#), “Creating and Attaching Cover Sheets”.

CSI

Places text on the cover sheet. This text usually is the general fax number for the enterprise. This command is valid only in cover sheets.

Syntax `{{csi string}}`

Example `{{csi 503-555-5481}}`

This example prints “503-555-5481” on the cover sheet.

{{Csi}} translates to the RightFax server field from_phone. For more information, [“Using Cover Sheets in a Broadcast”](#) on page 69.

For the procedure on using this command with cover sheets, see [Chapter 14, “Creating and Attaching Cover Sheets”](#).

For the procedure on setting default text for the CSI field, see [Chapter 9, “Setting Defaults for FCL Documents”](#).

DATE

Sets the month, day, and optionally the year in which to send the current document. {{Date}} can be combined with {{time}} to establish both a day and time to send a document.

The slashes between the month, day, and year are required. If no {{date}} command is specified, the Integration Module uses the date the FCL input file was received from the host. The year can be only two digits, and it can include the century. If the year is less than 96, the twenty-first century is assumed.

Syntax `{{date month/day/year}}`

Examples `{{date 11/2}}`
`{{date 11/20/2004}}`

The first example specifies that the current document is to be transmitted on November 2.

The second example specifies that the document is to be transmitted on November 20, 2004.

See also [“TIME”](#) on page 215

For the procedure that uses this command, see [Chapter 20, “Scheduling Document Transmission”](#).

DELAY

Schedules the document to be sent at a later time by the specified number of minutes.

Syntax `{{delay minutes}}`

Example `{{delay 10}}`

The example adds 10 minutes to the scheduled send time of the document.

See also [“BATCH”](#) on page 190
[“DATE”](#) on page 195
[“TIME”](#) on page 215

For the procedure that uses this command, see [Chapter 20, “Scheduling Document Transmission”](#).

DEPT

Specifies the department of the fax owner.

Syntax `{{dept department}}`

Example `{{dept sales}}`

The example specifies that the department associated with this fax is “Sales”.

For the procedure that uses this command, see [Chapter 21, “Setting Up Notification Messages of Document Transmission”](#).

EMAIL

Stores the sender’s e-mail address. This is usually used for routing a notification back to the sender that the document has been processed, has transmitted, or has failed to transmit

Syntax `{{email address}}`

Example `{{email user@acme.com}}`

The example stores the e-mail address user@acme.com for use in notifications.

For the procedure that uses this command, see [Chapter 21, “Setting Up Notification Messages of Document Transmission”](#).

EMPID

Specifies the employee ID of the fax owner.

Syntax `{{empid /D}}`

Example `{{empid 555-111-2222}}`

The example specifies that the employee ID associated with this fax is 555-111-2222.

For the procedure that uses this command, see [Chapter 21](#), “Setting Up Notification Messages of Document Transmission”.

END

Terminates the current document. This is a required command. This command is required in every FCL document. FCL documents may contain multiple `{{begin}}` and `{{end}}` commands, but the FCL between each set of commands will be rendered as a separate page.

Syntax `{{end}}`

Example `{{end}}`

See also “BEGIN” on [page 190](#)

For the procedure that uses this command, see [Chapter 8](#), “Creating FCL Documents”.

ENDCOVERTEXT

Ends the storing of cover sheet text and tells the RightFax Integration Module to again interpret incoming data.

Syntax `{{endcovertext}}`

Example `{{endcovertext}}`

For the procedure that uses this command, see [Chapter 14](#), “Creating and Attaching Cover Sheets”.

ENDCVT

Ends the conversion of an embedded foreign document that was begun with the `{{begin cvt}}` command.

Syntax `{{endcvt}}`

Example `{{endcvt}}`

See also “BEGINCVT” on [page 191](#)

ENDPOLY

Ends the current polygon, drawing a line from the current position to the first vertex of the polygon, using the current line width. `{{Endpoly}}` is used in conjunction with `{{startpoly}}`.

Syntax `{{endpoly}}`

Example `{{endpoly}}`

See also “LINEWIDTH” on [page 201](#)
“STARTPOLY” on [page 214](#)

For the procedure that uses this command, see [Chapter 15](#), “Creating Lines, Boxes, and Other Shapes”.

EXECUTE

Ends the current document, executes the commands to this point and begins a new document without clearing the previous image. Most of the attributes of the original document are preserved, except for the following, which are cleared: `{{altfax}}`, `{{company}}`, `{{contact}}`, `{{cover}}`, and `{{fax}}`.

If document processing is performed on a RightFax WorkServer, and if the FCL document contains a print command, then the `{{execute}}` command will force printing to occur from the WorkServer whether or not the Print service is enabled.

Syntax `{{execute}}`

Example `{{execute}}`

See also “END” on [page 196](#)
“FAX” on [page 197](#)

FAX

Assigns the fax number where the document will be sent. If this is not the first fax number specified for this document, the document is ended, and a new document begins with all the images of the original. The last page can be updated and new pages can be added, but prior pages will remain the same as in the original. By specifying multiple fax numbers, broadcasts to many different destinations can be made. If you have the InternetLink Module, then you can also insert an e-mail address into the fax command and documents will be e-mailed. For more information, see the *RightFax InternetLink Module Guide*.

Syntax `{{fax number}}`
`{{fax e-mail address}}`

Example `{{fax 555-555-5458}}`

The example assigns the fax number 555- 555-5458 to the document.

See also “TYPE” on [page 216](#)

`{{Fax}}` translates to the RightFax server field `to_faxnum`. For more information, “[Using Cover Sheets in a Broadcast](#)” on [page 69](#).

For the procedure that uses this command, see [Chapter 11](#), “[Specifying the Destination of the Document](#)”.

FF

Acts as a form feed in the FCL input file. It begins a new page at the current left and top margins, just as if an ASCII form feed character had come from the host application.

Syntax `{{ff}}`

Example `{{ff}}`

For the procedure that uses this command, see [Chapter 16](#), “[Setting Margins and Tabs](#)”.

FILE

Stores all subsequent input (until an `{{end}}` command) into the specified file name. You can specify the full path to the file. The default is `RightFax\Production\Include`. `{{File}}` differs from `{{list}}` in that it does not strip leading, white space, or blank lines.

Syntax `{{file name}}`

Example `{{file test.inc}}`

The example writes all subsequent input until an `{{end}}` command is encountered to the file `RightFax\Production\Include\test.inc`.

See also “LIST” on [page 201](#)

For the procedure that uses this command, see [Chapter 11](#), “[Specifying the Destination of the Document](#)”.

FILLBOX

Draws a box in the current line width at the specified coordinates, filling it with black or white (the default is black). The coordinates can be formatted (x,y) or (x y). The coordinates can be specified values, or you can specify a position relative to the current cursor position.

If only one coordinate pair is specified, the other coordinate pair is assumed to be the current position. Therefore, `{{fillbox (5,5)}}` is equivalent to `{{fillbox (x,y) (5,5)}}`. If both coordinate pairs are specified, you can specify a fill color. The color can be black or white.

Syntax `{{fillbox coord1 coord2 color}}`

Examples `{{fillbox (0,y) (7,7) white}}`
`{{fillbox 5 5 7 7}}`

The first example draws a box from the left side of the page at the current line to 7 units over, and 7 units down the current page, filling it in with white. This could be useful for covering up information after an `{{execute}}` command.

The second example draws a 2 by 2 unit filled black box at the coordinates (5,5) to (7,7) in current units.

See also [“BOX” on page 192](#)
[“LINEWIDTH” on page 201](#)
[“RBOX” on page 210](#)
[“RFILLBOX” on page 211](#)
[“UNITS” on page 217](#)

For the procedure that uses this command, see [Chapter 15](#), [“Creating Lines, Boxes, and Other Shapes”](#).

FONT

Changes the font currently in use and, optionally, the attributes of the font. The available font attributes are described in the following table.

Table B1 Font Attributes

Attribute	Description
Name	The name of the font. This can be one of the fonts installed with the Integration Module, a TrueType font, or any font supported by Microsoft Windows. Fonts are installed in <code>Rightfax\Production\Fonts</code> . The font name is used for all subsequent text until another <code>{{font}}</code> command appears in the document or until the end of the document.
Size	The size of a Windows font in points. The default is 12. The size can range from 3 to 288. This attribute is not available for the installed fonts.
Leading	The vertical spacing for the lines of text in points. This indicates the number of points to move down from the baseline of the current line to the baseline of the following line. A higher number increases the amount of vertical space used. One inch is 72 points. Thus, a leading of 12 points is 6 lines per inch (72 divided by 12 equals 6). You can enter leading in decimals.
Pitch	The horizontal spacing in characters per inch. A higher number decreases the amount of horizontal space used. A pitch of 10 is 10 characters per horizontal inch. You can enter pitch in decimals.
Weight	The weight of a Windows font. Weights are thin, extralight, light, regular, medium, semibold, bold, extrabold, and heavy. This attribute is not available for the installed fonts.

Table B1 Font Attributes (Continued)

Attribute	Description
Italic	Italicize a Windows font.
Strikeout	Draw a horizontal line through the text.

For more information on the installed fonts and support for TrueType fonts, see [Chapter 17, “Selecting and Configuring Fonts”](#).

Syntax `{{font name [size=##] [leading=##] [pitch=##] [weight] [italic] [strikeout]}}`

Examples `{{font courb12 leading=14 pitch=10}}`
`{{font timbi10}}`
`{{font “times new roman” size=24 extrabold italic}}`

The first example sets the current font to Courier Bold 12-point, with 14 points of leading and 10 characters per inch.

The second example sets the current font to Times Bold Italic 10-point, with default leading. Without specifying leading and pitch, this font will be proportional and vertical columns will not line up.

The third example uses the Microsoft Windows TrueType font Times New Roman, size 24, extra bold, and italic.

For the procedure that uses this command, see [Chapter 17, “Selecting and Configuring Fonts”](#).

FORM

Specifies a Class F TIF file to be overlaid on the current and subsequent pages. The specified form name can include the full path to the file or a path that is relative to RightFax\Production\Forms. The form is normally placed with the upper-left corner at (0,0) on the page. You can specify the location with x- and y-coordinates. Placement is relative to the upper-left corner of the current orientation, regardless of the current `{{placexy}}` value.

Alternatively, you can use this code to specify that no form should be overlaid.

Syntax `{{form name coord}}`

Examples `{{form order 0.5 0.2}}`
`{{form none}}`

The first example sets the form to RightFax\Production\Forms\order or

RightFax\Production\Forms\order.tif

and places it at (0.5, 0.2) on the page in current units.

The second example specifies that there is no form to be overlaid.

See also [“PLACE” on page 207](#)
[“UNITS” on page 217](#)

For the procedure that uses this command, see [Chapter 12, “Creating and Linking Background Forms”](#).

INCLUDE

Includes an FCL input file in the input stream. Input from an include file is interpreted just as if it came from the host application. Include files are useful when the same input is needed in many files. You can specify the full path to the file. The defaults is RightFax\Production\Include. If the include file cannot be found, the Integration Module tries again, adding .inc to the file name.

Syntax `{{include filename}}`

Example `{{include setup}}`

The example includes input from the file RightFax\Production\Include\setup. If that file does not exist, then it would include input from the file RightFax\Production\Include\setup.inc.

For the procedure that uses this command, see [Chapter 13, “Including the Same Files and Commands in Many Documents”](#).

LIBDOC

Attaches the specified RightFax library document.

Syntax `{{libdoc ID}}`

Example `{{libdoc InfoPack1}}`

The example attaches the library document called InfoPack1 to the end of the base document.

Library documents are frequently faxed documents (such as company literature, credit applications, and employment forms) that you create with FaxUtil. RightFax stores these in the RightFax\limage folder. For more information on library documents, see the *RightFax Administrator's Guide*.

For the procedure that uses this command, see [Chapter 19, "Attaching and Embedding Files, Signatures, and Graphics"](#).

LINE

Draws a line in the current line width on the current page at the specified coordinates in the current units. The coordinates can be formatted (x,y) or (x y). The coordinates can be specified values, or you can specify a position relative to the current cursor position. If both pairs of coordinates are not specified, `{{line}}` draws a horizontal line across the page under the baseline in the current font.

Syntax `{{line [coord1] [coord2]}}`

Examples `{{line (2,3) (4,5)}}`
`{{line}}`

The first example draws a line from coordinates (2,3) to (4,5) in current units.

The second example draws a line under the current font baseline.

See also ["LINETO" on page 200](#)
["LINEWIDTH" on page 201](#)
["RLINE" on page 211](#)
["RLINETO" on page 211](#)
["UNITS" on page 217](#)

For the procedure that uses this command, see [Chapter 15, "Creating Lines, Boxes, and Other Shapes"](#).

LINETO

Draws a line in the current line width on the current page from the current position to the specified coordinates in current units. The coordinates can be formatted (x,y) or (x y). The coordinates can be specified values, or you can specify a position relative to the current cursor position.

Syntax `{{lineto (coord)}}`

Example `{{lineto (3,5)}}`

The example draws a line from the current cursor position to (3,5) in current units.

See also ["LINETO" on page 200](#)
["LINEWIDTH" on page 201](#)
["RLINE" on page 211](#)
["RLINETO" on page 211](#)
["UNITS" on page 217](#)

For the procedure that uses this command, see [Chapter 15, "Creating Lines, Boxes, and Other Shapes"](#).

LINewidth

Sets the width to draw lines in points. A point is 1/72nd of an inch. A `{{linewidth}}` command with zero as the number of points indicates that no line is to be drawn. Any other line width will be rendered at least one pixel wide (approximately 1/200th of an inch). The default line width is one pixel (approximately 1/3 point).

Syntax `{{linewidth points}}`

Example `{{linewidth 1}}`

The example sets the line width to 1 point, or 1/72nd of an inch (about 3 pixels).

See also [“LINE” on page 200](#)

For the procedure that uses this command, see [Chapter 15](#), [“Creating Lines, Boxes, and Other Shapes”](#).

LIST

Writes subsequent input (until an `{{end}}` command) to the specified file name. You can specify the full path to the file. The default is `RightFax\Production\Include`. This process will replace a file of the same name. `{{List}}` differs from `{{file}}` only in that it will strip any leading white space or blank lines from the input. `{{List}}` is useful in creating and replacing broadcast lists.

Syntax `{{list filename}}`

Example `{{list bcast.inc}}`

The example writes subsequent input (until an `{{end}}` command) to `RightFax\Production\Include\bcast.inc`.

See also [“FILE” on page 197](#)

For the procedure that uses this command, see [Chapter 11](#), [“Specifying the Destination of the Document”](#).

LM

Sets the current left margin for rendering text, immediately changing the current position to reflect the new value. The width of the margin is specified in current units from the left edge of the page.

Syntax `{{lm margin}}`

Example `{{lm 0.5}}`

The example sets the left margin to 1/2 a unit from the left edge of the page, and moves to that location.

See also [“BM” on page 191](#)
[“TM” on page 215](#)
[“UNITS” on page 217](#)

For the procedure that uses this command, see [Chapter 16](#), [“Setting Margins and Tabs”](#).

LOOKUP

A lookup table can provide information that is not contained in the document that is sent from the host application, such as the recipient company name, fax number, e-mail address, and contact name.

The lookup FCL code uses the specified criteria to look up information in a table contained in a lookup file and then associate that information with other information.

Syntax `{{lookup criteria table}}`

Example `{{lookup AC123}}`

The example instructs the software to find the entry AC123 in a lookup table. A lookup table might look like this:

```
AC123{{company ACME Corp}}{{contact John Doe}}
OC456{{company Oregon Corp}}{{contact Jane Doe}}
Default{{company Arizona Corp}}{{contact Mary Doe}}
```

If you use `{{lookup AC123}}`, the software would insert “ACME Corp” and “John Doe” into the document. If you made an error, such as `{{lookup AC234}}` (which does not exist in the lookup table), then the software would insert the default information. If you do not specify the path to the lookup table, the default is `RightFax\Production\Include`.

LP, LPR, or PRINTER

Sets the default printer for the print FCL commands such as `{{type print}}` and `{{printnow}}`.

Syntax `{{lp|lpr|printer name}}`

Example `{{lp local}}`

The example sets the printer name to local.

See also “[PRINTNOW](#)” on [page 209](#)
“[TYPE](#)” on [page 216](#)

For the procedure that uses this command, see [Chapter 11](#), “[Specifying the Destination of the Document](#)”.

MOVETO

Changes the current position to the specified coordinates in units. The coordinates can be formatted (x,y) or (x y). The coordinates can be specified values, or you can specify a position relative to the current cursor position.

Syntax `{{moveto coord}}text`

Examples `{{moveto 3,4}}`Put this here.
`{{moveto x,5}}`Put this here.

The first example changes the position to (3,4) in current units.

The second example maintains the x position and moves vertically to 5 units down the page.

To specify a position relative to the current x- or y-coordinate, enter `x` or `y` in the command line. `X` is not a valid y-coordinate, and `Y` is not a valid x-coordinate.

See also “[POSITION](#)” on [page 209](#)
“[RMOVETO](#)” on [page 212](#)
“[UNITS](#)” on [page 217](#)

For the procedure that uses the `{{moveto}}` command, see [Chapter 15](#), “[Creating Lines, Boxes, and Other Shapes](#)”.

Using the `{{moveto}}` command in a Unix system

If you are using the `{{moveto}}` command in a Unix system that generates FCL documents, you must add a value to the Windows registry in order to support the command. On the RightFax server, edit the Windows registry. Navigate to the subkey `HKEY_LOCAL_MACHINE\Software\RightFax\Parse`. Add the `REG_MULTI_SIZE` registry value `UnixSpecials`. Enter the string `MoveTo` in the multi-string editor.

NOCOVER or NO_COVER

Same as `{{cover none}}` ([page 193](#)). No cover sheet is generated, even if you have defined a default cover sheet in the Integration Module Configuration program.

Syntax `{{nocover}}`

Example `{{nocover}}`

The example allows the document to be processed with no cover sheet.

For the procedure that uses this command, see [Chapter 14](#), “[Creating and Attaching Cover Sheets](#)”.

NOTE

Same as `{{REM}}` (page 210). Inserts any information; is commonly used for troubleshooting. It has no effect on the document.

Syntax `{{note text}}`

Example `{{note This came from the PO system}}`

The example inserts “This came from the PO system” into the FCL but not into the finished document.

NOTIFYHOST

Specifies the templates to be used to format the success or failure of the transmission of the document. It also can specify which ID to send the notifications. The template files (in the syntax below, these are `Successstemplate.inc` for successful documents and `Failuretemplate.inc` for failed documents) describe the notification that should be sent to the host application.

If a template name is not specified, then no notification will be sent. The channel specifies the notification channel that will send the notification message. The channel can be specified with an ID number (from 1 to 128) or the name assigned to the channel. If no channel is specified, the default channel will be used, number 16.

Syntax `{{notifyhost successstemplate failuretemplate
[channel]}}`

Example `{{notifyhost mysucc myfail mynotify}}`

The example shows that, if the document is sent successfully, the RightFax Integration Module generates a notification message using the `RightFax\Production\Include\Mysucc`

or

`RightFax\Production\Include\Mysucc.inc` notification templates. Similarly, the `Myfail` template will be used if the fax is not successfully sent. Notifications spool to the `Mynotify` ID.

See also “[ONERROR](#)” on page 203
“[ONSUCCESS](#)” on page 205

For the procedure that uses this command, see [Chapter 21](#), “[Setting Up Notification Messages of Document Transmission](#)”.

ONERROR

Describes what the RightFax Integration Module does in the event the document fails during transmission. This command overrides the settings you establish in the Integration Module Configuration program (see “[Setting the default “success” action](#)” on page 135) for each document in which you use the `{{onerror}}` command.

Options for `{{onerror}}` are fax, delete, email (or mime), certified, nothing, or print.

Table B2 Onerror Options

Option	Description
Certified	<p>If a transmission error occurs in faxing, the document will be sent as a certified e-mail document. This option requires that you have licensed the SecureDocs Module.</p> <p>To configure a notification to alert you that a document failed as a fax and was sent as a certified e-mail, see Chapter 21, "Setting Up Notification Messages of Document Transmission".</p>
Delete	<p>If a transmission error occurs in faxing, the fax image will be deleted. If you use <code>{{onerror delete}}</code>, then successful faxes also will be deleted.</p>
Email or mime	<p>If a transmission error occurs in faxing, the document will be sent via e-mail. This option requires that you have licensed the InternetLink Module.</p> <p>To configure a notification to alert you that a document failed as a fax and was sent as an e-mail, see Chapter 21, "Setting Up Notification Messages of Document Transmission".</p>
Fax	<p>If a transmission error occurs in faxing, the document will be sent via fax to another fax number. Specify the fax number and whether to delete the fax image after transmission.. The fax number you enter must be contiguous (no spaces or tabs).</p> <p>If you don't enter a fax number, the software uses the default number specified in the Integration Module Configuration program (see Chapter 9, "Setting Defaults for FCL Documents").</p>

Table B2 Onerror Options (Continued)

Option	Description
Nothing	<p>If a transmission error occurs in faxing, no special action is taken. This overrides any defaults you set with the Integration Module Configuration program (see "Setting the default "success" action" on page 135).</p>
Print	<p>If a transmission error occurs in faxing, the document will print. Enter a printer ID defined in Enterprise Fax Manager.</p> <p>If you don't enter a printer, the software uses the default printer specified in the Integration Module Configuration program (see Chapter 9, "Setting Defaults for FCL Documents").</p>

Syntax `{{onerror nothing}}`
`{{onerror delete}}`
`{{onerror fax number delete}}`
`{{onerror email|mime|certified address delete}}`

Examples `{{onerror fax 503-555-1234 delete}}`
`{{onerror email css@oswego.com}}`
`{{onerror certified css@oswego.com}}`

The first example shows that, upon failed transmission, all pages will be faxed to 503-555-1234 and then deleted.

The second example shows that, upon failed transmission, the document will be e-mailed to `css@oswego.com` via the InternetLink Module.

The third example shows that, upon failed transmission, the document will be e-mailed to `css@oswego.com` as a certified document via the SecureDocs Module.

For more information on the InternetLink Module, see the *RightFax InternetLink Module Guide*. For more information on the SecureDocs Module, see the *RightFax SecureDocs Module Guide*.

See also [“NOTIFYHOST” on page 203](#)
[“ONSUCCESS” on page 205](#)

{{Onerror delete}} translates to the RightFax server field faxflag_autodeleteall. For more information, [“Using Cover Sheets in a Broadcast” on page 69](#).

For the procedure that uses this command, see [Chapter 23, “Setting Up Actions on Document Transmission”](#).

ONSUCCESS

Describes what the RightFax Integration Module does in the event the document transmits successfully. This command overrides the settings you establish in the Integration Module Configuration program([“Setting the default “success” action” on page 135](#)) for each document in which you use the {{onsuccess}} command.

Options for `{{onsuccess}}` are fax, delete, email (or mime), certified, nothing, or print.

Table B3 Onerror Options

Option	Description
Certified	<p>When a transmission succeeds in faxing, the document will also be sent as a certified e-mail document. This option requires that you have licensed the SecureDocs Module.</p> <p>To configure a notification to alert you that a document was sent as a certified e-mail, see Chapter 21, "Setting Up Notification Messages of Document Transmission".</p>
Delete	<p>When a transmission succeeds in faxing, the fax image will be deleted.</p>
Email or mime	<p>When a transmission succeeds in faxing, the document will also be sent via e-mail. This option requires that you have licensed the InternetLink Module.</p> <p>To configure a notification to alert you that a document was sent as an e-mail, see Chapter 21, "Setting Up Notification Messages of Document Transmission".</p>
Fax	<p>When a transmission succeeds in faxing, the document will be sent via fax to another fax number. Specify the fax number and whether to delete the fax image after transmission.. The fax number you enter must be contiguous (no spaces or tabs).</p> <p>If you don't enter a fax number, the software uses the default number specified in the Integration Module Configuration program (see Chapter 9, "Setting Defaults for FCL Documents").</p>
Nothing	<p>When a transmission succeeds in faxing, no special action is taken. This overrides any defaults you set with the Integration Module Configuration program (see "Setting the default "success" action" on page 135).</p>

Table B3 Onerror Options (Continued)

Option	Description
Print	<p>When a transmission succeeds in faxing, the document will print. Enter a printer ID defined in Enterprise Fax Manager.</p> <p>If you don't enter a printer, the software uses the default printer specified in the Integration Module Configuration program (see Chapter 9, "Setting Defaults for FCL Documents")</p>

Syntax `{{onsuccess nothing}}`
`{{onsuccess delete}}`
`{{onsuccess fax number delete}}`
`{{onsuccess email|mime|certified address delete}}`

Examples `{{onsuccess fax 503-555-1234 delete}}`
`{{onsuccess email css@oswego.com}}`
`{{onsuccess certified css@oswego.com}}`

The first example shows that, upon successful transmission, all pages of the document are faxed to 503-555-1234 and then deleted.

The second example shows that, after successful fax transmission, the document will be e-mailed to `css@oswego.com` via the InternetLink Module.

The third example shows that, after successful fax transmission, the document will be e-mailed to `css@oswego.com` as a certified document via the SecureDocs Module.

For more information on the InternetLink Module, see the *RightFax InternetLink Module Guide*. For more information on the SecureDocs Module, see the *RightFax SecureDocs Module Guide*.

See also ["NOTIFYHOST" on page 203](#)
["ONERROR" on page 203](#)

{{Onsuccess delete}} translates to the RightFax server field faxflag_autodelete. For more information, “[Using Cover Sheets in a Broadcast](#)” on [page 69](#).

For the procedure that uses this command, see [Chapter 23](#), “[Setting Up Actions on Document Transmission](#)”.

ORIENT

Sets the page orientation for subsequent text and moves the current position to (0,0) on the page. Text specified prior to the {{orient}} command will be rendered in the prior page orientation.

Syntax {{orient {portrait|landscape}}}

Example {{orient portrait}}

The example sets the page orientation to portrait for subsequent text and graphic blocks.

For the procedure that uses this command, see [Chapter 18](#), “[Setting Page Orientation and Image Quality](#)”.

OWNER

Specifies the document owner's name.

Syntax {{owner name}}

Example {{owner Jane Doe}}

The example specifies the owner of this document as “Jane Doe”.

{{Owner}} translates to the RightFax server field from_name. For more information, “[Using Cover Sheets in a Broadcast](#)” on [page 69](#).

For the procedure that uses this command, see [Chapter 14](#), “[Creating and Attaching Cover Sheets](#)”.

PLACE

Positions the specified Class F TIF image on the page. The file can be specified using the full path or relative to RightFax\Production\Forms. You can specify the placement of the image using x- and y-coordinates. If no coordinates are specified, the graphic is placed at the current cursor location.

By default, the graphic is placed in the upper-left corner. The placement can be specified with the {{placexy}} command. {{Place}} is also affected by the current page orientation and measurement units.

Syntax {{place graphic coord}}

Example {{place yoyodyne.tif 5 4}}

The example places the graphic image Yoyodyne.tif on the page at (5,4) in specified units.

See also “[PLACEXY](#)” on [page 208](#)
“[PLACEALL](#)” on [page 207](#)

For the procedure that uses this command, see [Chapter 19](#), “[Attaching and Embedding Files, Signatures, and Graphics](#)”.

PLACEALL

Places the specified graphic image on the current and all subsequent FCL pages (but not on file attachments). The file can be specified using the full path name or a path relative to

RightFax\Production\Forms. Optionally, x- and y-coordinates can be specified in current units. If no coordinates are specified, the graphic is placed at the current x and y location.

You can use multiple `{{placeall}}` commands.

By default, the graphic block is placed in the upper-left corner. The position can be specified with the `{{placexy}}` command. `{{Placeall}}` is also affected by the current page orientations and measurement units.

Syntax `{{placeall graphic x y}}`

Example `{{placeall yoyodyne.tif 5 4}}`

See also “PLACE” on page 207
“PLACEXY” on page 208

The example places the file Yoyodyne.tif in the current and all subsequent pages at the location 5 4 (in units, such as inches or centimeters, that you set in the Integration Module Configuration program—see Chapter 9, “Setting Defaults for FCL Documents”).

For the procedure that uses this command, see Chapter 19, “Attaching and Embedding Files, Signatures, and Graphics”.

PLACELAST

Places the specified graphic image on the last page. The file can be specified using the full path name or a path relative to RightFax\Production\Forms. Optionally, x- and y-coordinates can be specified in current units. If no coordinates are specified, the graphic is placed at the current x and y location.

You can issue multiple `{{placelast}}` commands.

By default, the graphic block is placed in the upper-left corner. The placement can be specified with the `{{placexy}}` command. `{{Placeall}}` is also affected by the current page orientations and measurement units.

Syntax `{{placelast graphic x y}}`

Example `{{placelast yoyodyne.tif 5 4}}`

See also “PLACE” on page 207
“PLACEXY” on page 208

The example places the file Yoyodyne.tif in the last page at the location 5 4 (in units, such as inches or centimeters, that you set in the Integration Module Configuration program—see Chapter 9, “Setting Defaults for FCL Documents”).

For the procedure that uses this command, see Chapter 19, “Attaching and Embedding Files, Signatures, and Graphics”.

PLACEXY

Specifies how graphic images and box text should be placed on the page. Horizontal placement can be left, center, or right. Vertical placement can be top, center, or bottom. When placing graphic images, `{{placexy}}` denotes the placement of the graphic relative to the current position of the cursor. When adding text in a box, `{{placexy}}` describes the location of the text in the box. If no `{{placexy}}` command is specified, left and top are used.

Syntax `{{placexy horizontal vertical}}`

Example `{{placexy center center}}`

The example instructs the RightFax Integration Module to interpret coordinates in subsequent `{{place}}` commands as the horizontal and vertical center of the graphic block and to horizontally and vertically center text in subsequent `{{box}}` or `{{rbox}}` commands.

See also “PLACE” on page 207

For the procedure that uses this command, see Chapter 19, “Attaching and Embedding Files, Signatures, and Graphics”.

POSITION

Changes the current location on the page according to the currently selected font. The coordinates can be formatted (x,y) or (x y).

Syntax `{{position coord}}`

Example `{{position 5, 20}}`

The example moves the position to the fifth column of the twentieth row of the page (using the specified font as the guide to the character size).

See also [“MOVETO” on page 202](#)
[“RMOVETO” on page 212](#)

For the procedure that uses this command, see [Chapter 15](#), [“Creating Lines, Boxes, and Other Shapes”](#).

PREVIEW

Holds the document for preview in the FaxUtil mailbox.

Syntax `{{preview}}`

Example `{{preview}}`

For the procedure that uses this command, see [Chapter 20](#), [“Scheduling Document Transmission”](#).

PRINTER

See [“LP, LPR, or PRINTER” on page 202](#)

PRINTNOW

Prints a copy of the current document immediately. You can specify the number of copies to print. If the number of copies is not specified, the default is one copy.

Syntax `{{printnow print copies}}`

Example `{{printnow print 2}}`

The example immediately prints two copies of the document.

See also [“TYPE” on page 216](#)
[“LP, LPR, or PRINTER” on page 202](#)

PRIORITY

Specifies the priority at which the document is to be processed and scheduled. High priority documents will be processed and sent before low priority documents of the same scheduled time. If no `{{priority}}` command is specified, low priority is assumed.

Priority can be 0, 1, or 2, representing low, medium, or high. Normal is the same as medium. If you do not insert a priority command, the default is low (0) priority.

Syntax `{{priority {low|0|medium|normal|1|high|2}}}`

Example `{{priority high}}`
`{{priority 2}}`

Both examples specify a high priority, because “2” is equivalent to “high.”

See also [“DATE” on page 195](#)
[“TIME” on page 215](#)

`{{Priority}}` translates to the RightFax server field `ucPriority`. For more information, see [“Using Cover Sheets in a Broadcast” on page 69](#).

For the procedure that uses this command, see [Chapter 20](#), [“Scheduling Document Transmission”](#).

QUALITY

Specifies the fax resolution at which the document will send. Standard resolution is 204 x 98 dots-per-inch (dpi). Fine resolution is 204 x 196 dpi. `{{Quality}}` only affects the vertical resolution. `{{Quality}}` should be set on the first page of a document and maintained throughout the document. Otherwise, pages will stretch or compress when sent. If no `{{quality}}` command is specified, the default transmission quality in Integration Module Configuration program is used.

Syntax `{{quality {standard|fine}}}`

Example `{{quality fine}}`

The example sets the fax quality for the document to fine (204 x 196 dpi).

`{{Quality}}` translates to the RightFax server field `finemode`. For more information, see [“Using Cover Sheets in a Broadcast”](#) on page 69.

For the procedure that uses this command, see [Chapter 18](#), [“Setting Page Orientation and Image Quality”](#).

RBOX

Draws a box in the current line width at the specified coordinates, optionally filling it with the specified text. The coordinates can be formatted (x,y) or (x y). The coordinates can be specified values, or you can specify a position relative to the current cursor position. If only one coordinate pair is specified, the other coordinate pair is assumed to be the current position. Therefore, `{{rbox (5,5)}}` is equivalent to `{{rbox (0,0) (5,5)}}`. If both coordinate pairs are specified, you can embed a line of text. The text must be

surrounded by quotation marks and is limited to one line. The text is placed in the box according to the current `{{placexy}}` settings, which default to the upper-left corner.

Syntax `{{rbox coord1 coord2 “text”}}`

Examples `{{rbox (0,3) (2,4) “Hi There”}}`
`{{rbox -1 -1 1 1}}`

The first example draws a box 3 units down from the current position to 2 units across and 4 units down, placing the text “Hi There” within it.

The second example draws a 2-by-2-unit box one unit back and one unit up from the current position, centering it on the current position.

See also [“BOX”](#) on page 192

For the procedure that uses this command, see [Chapter 15](#), [“Creating Lines, Boxes, and Other Shapes”](#).

REM

Same as [“NOTE”](#). Inserts any information; is commonly used for troubleshooting. It has no effect on the document.

Syntax `{{rem text}}`

Example `{{rem This came from the PO system}}`

The example inserts “This came from the PO system” into the FCL, but not into the finished document.

REPLYTO or REPLY_TO

Specifies a recipient for a notification. You can request that an HTTP post be sent back to the host as a notification when you use the RightFax XML Interface. REPLYTO is the field in the submit post that the XML Interface populates to determine where to send the notification.

For more information on the RightFax XML Interface, see [Chapter 25, “Programming for the RightFax API for Java”](#).

Syntax `{{replyto recipient}}`

Example `{{replyto www.captaris.com}}`

The example specifies that the XML notification should return to `www.captaris.com`.

For the procedure that uses this command, see [Chapter 21, “Setting Up Notification Messages of Document Transmission”](#).

RFILLBOX

Draws a box in the current line width at the specified relative coordinates, filling it black or white (black if none is specified). The coordinates can be formatted (x,y) or (x y). The coordinates can be specified values, or you can specify a position relative to the current cursor position. If only one coordinate pair is specified, the other coordinate pair is assumed to be the current position. Therefore, `{{rfillbox (5,5)}}` is equivalent to `{{rfillbox (0,0) (5,5)}}`. If both coordinate pairs are specified, you can specify a fill color. The color can be black or white.

Syntax `{{rfillbox coord1 coord2 color}}`

Example `{{rfillbox (3,3) white}}`
`{{rfillbox -1 -1 1 1}}`

The first example draws a box from the current position to 3 units over and 3 units down from the current position, filling it in with white. This could be useful for covering up information after an `{{execute}}` command.

The second example draws a 2-by-2-unit filled black box at coordinates (-1,-1) to (1,1) relative to the current position in current units.

See also [“BOX” on page 192](#)
[“FILLBOX” on page 197](#)

For the procedure that uses this command, see [Chapter 15, “Creating Lines, Boxes, and Other Shapes”](#).

RLINE

Draws a line in the current line width on the current page at the specified relative coordinates in current units. The coordinates can be formatted (x,y) or (x y). The coordinates can be specified values, or you can specify a position relative to the current cursor position. If either coordinate pair is not specified, `{{rline}}` draws a horizontal line across the page just under the baseline of the current font.

Syntax `{{rline [(coord1) (coord2)]}}`

Examples `{{rline (2,3) (4,5)}}`
`{{rline}}`

The first example draws a line from coordinates (2,3) to (4,5) relative to the current position in the current units.

The second example draws a line under the current font baseline.

See also [“LINE” on page 200](#)
[“UNITS” on page 217](#)

For the procedure that uses this command, see [Chapter 15, “Creating Lines, Boxes, and Other Shapes”](#).

RLINETO

Draws a line in the current line width on the current page from the current position to the specified relative coordinates in current units. The coordinates can be formatted (x,y) or (x y). The position is relative to the current cursor position in the current units.

Syntax `{{rlineto (coord)}}`

Example `{{rlineto (3,5)}}`

The example draws a line from the current cursor position to (3,5) in current units.

See also [“LINE” on page 200](#)
[“LINETO” on page 200](#)
[“LINEWIDTH” on page 201](#)
[“RLINE” on page 211](#)
[“UNITS” on page 217](#)

For the procedure that uses this command, see [Chapter 15](#), [“Creating Lines, Boxes, and Other Shapes”](#).

RMOVETO

Changes the current cursor position to the specified relative coordinates in the specified units. The coordinates can be formatted (x,y) or (x y).

Syntax `{{rmoveto coord}}`

Example `{{rmoveto 3,4}}`

The example changes the position to (3,4) from the current position in the specified units.

See also [“MOVETO” on page 202](#)
[“POSITION” on page 209](#)
[“UNITS” on page 217](#)

For the procedure that uses this command, see [Chapter 15](#), [“Creating Lines, Boxes, and Other Shapes”](#).

RSTARTPOLY

Moves the current cursor position to the relative coordinate specified and starts a polygon. The coordinates can be formatted (x,y) or (x y). This command is different from `{{startpoly}}` on [page 214](#). The command `{{startpoly 4,5}}` would begin a polygon at a point 4,5 units from the upper-left corner of the fax. The command `{{rstartpoly 4,5}}` would begin a polygon at a point 4,5 units from the current cursor location.

By itself, `{{rstartpoly}}` does not create a polygon; it establishes the starting point. Without the two `{{lineto}}` commands that create the lines of the polygon (see example), `{{rstartpoly}}` creates nothing. `{{endpoly}}` closes the polygon by connecting the lines created by the `{{lineto}}` commands. If you do not specify coordinates, the polygon starts at the position in the document where the command appears.

Syntax `{{rstartpoly coord}}`

Example `{{rstartpoly 0,1}} {{lineto 7,y}} {{lineto 7,10}} {{endpoly}}`

The example uses `{{rstartpoly}}` to begin a polygon at a point 0,1 units, relative to the current cursor location. (You establish units, such as inches or centimeters, in the Integration Module Configuration program—see [Chapter 9](#), [“Setting Defaults for FCL Documents”](#)).

See also [“ENDPOLY” on page 196](#)
[“LINETO” on page 200](#)
[“RLINETO” on page 211](#)
[“STARTPOLY” on page 214](#)
[“UNITS” on page 217](#)

For the procedure that uses this command, see [Chapter 15](#), [“Creating Lines, Boxes, and Other Shapes”](#).

RTI

Places text on the cover sheet. This usually is the name of the sending company. This command is valid only in cover sheets.

Syntax `{{rti string}}`

Example `{{rti ABC Company}}`

This example prints “ABC Company” on the cover sheet.

For the procedure on using this command with cover sheets, see [Chapter 14](#), [“Creating and Attaching Cover Sheets”](#).

For the procedure on setting a default RTI, see [Chapter 9](#), [“Setting Defaults for FCL Documents”](#).

SETTAB

Creates a tab stop specified by identifiers in the command. The identifiers are:

- Any whole number starting with zero to identify a tab group. You can specify up to 20 tabs in a document, numbered 0 through 19.
- Any measurement to define the size (in inches) of the tab.
- Alignment of the tab (C for center, L for left, R for right, D for aligning decimals in a group of numbers).

Syntax `{{settab tab coord {l|r|c|d}}}`

Examples `{{settab 0 1.5 L}}`
`{{settab 1 2.5 C}}`

The first example creates a global tab stop labeled group 0 at 1.5 inches, aligned left.

The second example creates a global tab stop labeled group 1 at 2.5 inches, aligned center.

See also “[CLEARTABS](#)” on [page 192](#)
“[TAB](#)” on [page 214](#)

For the procedure that uses this command, see [Chapter 16](#), “[Setting Margins and Tabs](#)”.

SIGN, SIGNED, or @

Places the file specified by the `{{signature}}` command after you have predefined a file name for `{{signature}}`. These three commands insert a signature in the document.

After you have predefined `{{signature}}`, you can insert one of these three commands to insert the signature in the document.

Syntax `{{{sign|signed|@}}}`

Example `{{sign}}`

See also “[SIGNATURE](#)” on [page 213](#)

For the procedure that uses this command, see [Chapter 19](#), “[Attaching and Embedding Files, Signatures, and Graphics](#)”.

SIGNATURE

Specifies the name of a graphic file of a signature that should appear in the document. The signature must be created and saved as a graphic file.

Syntax `{{signature filename}}`

Example `{{signature FredJones.tif}}`

See also “[SIGN, SIGNED, or @](#)” on [page 213](#)

For the procedure that uses this command, see [Chapter 19](#), “[Attaching and Embedding Files, Signatures, and Graphics](#)”.

SMS

Specifies the phone number of the SMS-capable device that will receive notifications about the fax transmission

Syntax `{{sms pagerID}}`

Example `{{sms 520-555-1212}}`

See also “[TYPE](#)” (SMS) on [page 216](#) and “[SMSMSG](#)” on [page 213](#)

For the procedure that uses this command, see [Chapter 21](#), “[Setting Up Notification Messages of Document Transmission](#)”.

SMSMSG

Specifies the text of the SMS notification to send

Syntax `{{smsmsg text}}`

Example `{{smsmsg Fax was sent successfully}}`

See also “[TYPE](#)” (SMS) on [page 216](#) and “[SMS](#)” (SMS) on [page 213](#)

For the procedure that uses this command, see [Chapter 21](#), “Setting Up Notification Messages of Document Transmission”.

STARTPOLY

Moves the current position to the coordinates specified and starts a polygon. The coordinates can be formatted (x,y) or (x y). The coordinates can be specified values, or you can specify a position relative to the current cursor position.

{{Startpoly}} is different from {{rstartpoly}} ([page 212](#)).

- {{Startpoly}} starts a polygon at coordinates that are relative to the upper-left corner of the document. (You establish units, such as inches or centimeters, in the Integration Module Configuration program—see [Chapter 9](#), “Setting Defaults for FCL Documents”).
- {{Rstartpoly}} starts a polygon at coordinates that are relative to the location that the command appears in the document.

By itself, {{startpoly}} does not create a polygon; it establishes the starting point. Without the two {{lineto}} commands that create the lines of the polygon (see example), {{startpoly}} creates nothing. {{Endpoly}} closes the polygon by connecting the lines created by the {{lineto}} commands. If you do not specify coordinates, the polygon starts at the position in the document where the command appears.

Subsequent calls to {{lineto}} or {{rlineto}} specify the vertices of the polygon, and {{endpoly}} is used to close the polygon. If no coordinates are specified, the current position is used for the start of the polygon.

Syntax `{{startpoly coord}}`

Example `{{startpoly (3,4)}}{{lineto 7,y}}{{lineto 7,10}}{{endpoly}}`

The example starts a polygon at (3,4) in current units.

See also “ENDPOLY” on [page 196](#)
 “RSTARTPOLY” on [page 212](#)
 “UNITS” on [page 217](#)

For the procedure that uses this command, see [Chapter 15](#), “Creating Lines, Boxes, and Other Shapes”.

TAB

Creates a single tab stop based on the information you create with the {{settab}} command.

Syntax `{{tab tab stop text|number}}`

Examples
`{{tab 3 1048.01}}`
`{{tab 3 16.8575}}`
`{{tab 1 Hello world}}`

The first two examples are of tab stops for a tab labeled group 3 (you defined this tab group with the {{settab}} command). For these examples, if you defined group 3 tabs as {{settab 3 4.5 D}}, then the numbers 1048.01 and 16.8575 would appear with their decimal points aligned vertically at 4.5 inches from the left margin.

The third example is of a tab stop for a tab labeled group 1. If you defined group 1 tabs as {{settab 1 1.5 L}}, then the words “Hello world” would appear aligned left at 1.5 inches from the left margin.

See also “CLEARTABS” on [page 192](#)
 “SETTAB” on [page 213](#)

For the procedure that uses this command, see [Chapter 16](#), “Setting Margins and Tabs”.

TERMID

Specifies the terminal identification from which the document originated.

Syntax `{{termid ID}}`

Example `{{termid A3}}`

The example specifies that the current document came from terminal A3.

For the procedure that uses this command, see [Chapter 21](#), “Setting Up Notification Messages of Document Transmission”.

TIME

Sets the time today when the document should be transmitted. If you enter a time that is earlier than the current time, the document will be sent immediately. You can combine the `{{time}}` with `{{date}}` commands to schedule the document.

The colon is required if hours and minutes are specified. If no `{{time}}` command is specified, the time at which the document was received by the RightFax Integration Module is used.

If you enter a time that is in the past, the document is transmitted immediately. Documents are also transmitted immediately with the time 0.

Syntax `{{time hour:minute}}`

Example `{{time 22:45}}`

The example sets the time to transmit the current document to 10:45 P.M. today

See also “DATE” on [page 195](#)

.For the procedure that uses this command, see [Chapter 20](#), “Scheduling Document Transmission”.

TM

Sets the top margin for the current and subsequent pages in the specified units. Text will not render above this margin on the page after this command. By default, there is no top margin (in other words, the top margin is zero).

Syntax `{{tm margin}}`

Example `{{tm 0.25}}`

The example sets the top margin for the current document to 1/4 inch (or current unit).

See also “BM” on [page 191](#)
“LM” on [page 201](#)

For the procedure that uses this command, see [Chapter 16](#), “Setting Margins and Tabs”.

TRANID

Sets the identification of the transaction that produced the document.

Syntax `{{trnid /D}}`

Example `{{trnid BR549}}`

The example sets the transaction ID for this document to “BR549”.

For the procedure that uses this command, see [Chapter 21](#), “Setting Up Notification Messages of Document Transmission”.

TYPE

Specifies the type of document. The available document types are listed here.

Table B4 Type Options

Option	Description
Fax	Documents are rendered as TIF images and transmitted via fax.
Print	Documents are rendered as TIF images, scaled to full-size, and then printed. You can specify the number of copies to print. If the number of copies is not specified, one copy prints. The printed document will include production cover sheets (.cov files). Enterprise cover sheets (.pcl and .doc files) are not printed.
File	Documents are rendered as TIF images and placed in the folder that you specify. This command cannot be used when submitting Embedded or False First Page (FFP) documents to the Integration Module.
Email	Documents are rendered as text and included in the body of an e-mail message. To send documents as e-mail, you must license the InternetLink Module. For more information, see the <i>RightFax InternetLink Module Guide</i> .
Mime	Documents are rendered as TIF images and included as an attachment to an e-mail message. To send documents as e-mail, you must license the RightFax InternetLink Module. For more information, see the <i>RightFax InternetLink Module Guide</i> .

Table B4 Type Options (Continued)

Option	Description
Certified	Documents are sent as certified e-mails. To send documents certified delivery, you must license and install the RightFax SecureDocs Module. For more information, see the <i>RightFax SecureDocs Module Guide</i> .
SMS	Specifies that notifications about the fax transmission will be sent to an SMS-capable device. Use the syntax {{type sms serviceID}} where <i>serviceID</i> is the SMS/Pager service ID for an SMS service defined in Enterprise Fax Manager. For more information on creating Pager/SMS services in Enterprise Fax Manager, refer to the <i>RightFax Administrator's Guide</i> . For more information on sending SMS notifications, see Chapter 21, "Setting Up Notification Messages of Document Transmission" .

Syntax **{{type type copies filename}}**

Examples **{{type fax}}**
{{type file c:\Program
Files\RightFax\Production\Forms\Example.tif}}
{{type print 2}}

The first example establishes that the document will be sent as a fax, rather than a file or be sent to a printer. This is the Integration Module default—all documents are assumed to be faxes. You would use **{{type fax}}** only if you had made file or print the default, but wanted to fax a particular document or group of documents. For information on how you can change the default type from fax to either file or print, see [Chapter 9, "Setting Defaults for FCL Documents"](#).

The second example creates a document called Example.tif in the Forms folder.

The third example prints two copies of the document. The printer used is the default established in the Integration Module Configuration program (it must also be defined in Enterprise Fax Manager). To change the printer with FCL, you can include `{{lp}}`, `{{lpr}}`, or `{{printer}}` (page 202). Or, you can change the default printer in Enterprise Fax Manager.

For information on using the Integration Module Configuration program to establish a default printer, see Chapter 9, “Setting Defaults for FCL Documents”.

See also “PRINTNOW” on page 209
“LP, LPR, or PRINTER” on page 202

For the procedure that uses this command, see Chapter 11 “Specifying the Destination of the Document”.

UNDERLINE

Sets underlining on or off for subsequent text in the current document.

Syntax `{{underline {on|off}}}`

Example `{{underline on}}`

The example turns on underlining.

See also “FONT” on page 198

For the procedure that uses this command, see Chapter 17, “Selecting and Configuring Fonts”.

UNIQUEID or UNIQUE_ID

Provides a tracking mechanism in FaxUtil by setting the ID field for one destination (one fax number) within the document. (For information on FaxUtil, see the *RightFax Administrator’s Guide*.)

You can use up to 15 alphanumeric characters in the `{{unique_id}}` command. If you use more than 15 characters, the command is truncated at the 15th character.

You can also use the `{{unique_id}}` command in cover sheet creation and notification templates. If you do not use this command in a document, the ID default is `prod docnum`, where `docnum` is a unique integer.

Syntax `{{unique_id /ID}}`

Examples`{{unique_id test:01ea}}`
`{{unique_id test:01eb}}`

`{{Unique_id}}` translates to the RightFax server field `unique_id`.

For the procedure that uses this command, see Chapter 21, “Setting Up Notification Messages of Document Transmission”.

UNITS

Sets the units of measurement to use for subsequent commands in the current document. Units of measure can be:

- Inches – in
- Centimeters – cm
- Points (72nds of an inch) –points
- Pixels (200ths of an inch) –pixels

The default is inches. You set the default unit of measurement for all documents in the FCL processor settings in the Integration Module Configuration program (see Chapter 9, “Setting Defaults for FCL Documents”). The `{{units}}` command overrides this global default for each document in which the command is used.

Syntax `{{units measure}}`

Example `{{units cm}}`

The example sets the units of measurement for subsequent commands in this document to be centimeters.

For the procedure that uses this command, see Chapter 9, “Setting Defaults for FCL Documents”.

USER1, USER2, or USER3

These commands can hold user-defined information.

These commands can be used only in notifications and Integration Module (.cov) cover sheets.

Syntax `{{user1 user information}}`
`{{user2 user information}}`
`{{user3 user information}}`

Example `{{user1 Some important information}}`

For the procedure that uses this command, see [Chapter 21](#), “Setting Up Notification Messages of Document Transmission”.

USERID

Identifies the creator of this document.

Syntax `{{userid ID}}`

Example `{{userid John Doe}}`

The example sets the user ID for this document to “John Doe”.

For the procedure that uses this command, see [Chapter 21](#), “Setting Up Notification Messages of Document Transmission”.

UTC

To avoid confusion caused by different time zones, the UTC command sets the date and time when a document should be sent in Universal Coordinated Time.

Alternatively, you can use a Time Zone Designator (TZD). A TZD of *+hh:mm* or *-hh:mm* indicates that the date/time uses a local time zone that is a particular number of hours and minutes (specified by *hh* and *mm*) ahead of or behind Universal Coordinated Time.

If you do not specify a TZD, then Universal Coordinated Time is assumed.

Syntax `{{UTC YYYY{-/}MM{-/}DD{T|}hh:mm:ss TZD}}`

Examples `{{UTC 2000-01-24T22:25:00Z}}`
`{{UTC 2000-01-24T23:25Z+01:00}}`
`{{UTC 2000-01-24T14:25-08:00}}`

Times must be represented as two digits, with leading zeros as necessary, with the exception of the four-digit year and the TZD.

Table B5 Time Options

Option	Definition	Example
YYYY	Year	1970-2038
MM	Month	01-12 (January = 01)
DD	Day	01-31
hh	Hour	00-23
mm	Minute	00-59
ss	Second	00-59
TZD	Time zone designator (Z or +hh:mm or -hh:mm)	

For the procedure that uses this command, see [Chapter 20](#), “Scheduling Document Transmission”.

VOICE

Sets the voice number to be associated with the current document. This is useful for specifying a voice number on a cover sheet or notification.

Syntax `{{voice number}}`

Example `{{voice 503-555-4329}}`

The example sets the voice number associated with this document to (503) 555-4329.

{{Voice}} translates to the RightFax server field to_contactnum. For more information, see [“Using Cover Sheets in a Broadcast”](#) on page 69.

For the procedure that uses this command, see [Chapter 21](#), [“Setting Up Notification Messages of Document Transmission”](#).

WINSECID

Specifies the RightFax user; or, if you have not created the user yet in Enterprise Fax Manager, {{winsecid}} creates it. Because {{winsecid}} can create a user, it is different from [“USERID”](#) (page 218), which is informational.

Captaris recommends that you have a {{winsecid}} command in the global.beg file. If you do not (and you do not have a {{winsecid}} command in the FCL from the host), then the default user account will be used to send documents from the Integration Module. The settings for a default user are typically not optimized for sending documents from the Integration Module (notifications, cover sheet settings, etc.).

By changing the {{winsecid}} command in the global.beg file, you can specify or change the default RightFax user account used by the Integration Module. (Open the global.beg file using Notepad or another text editor and change the user name in {{winsecid}}.)

Syntax {{winsecid *user*}}

Example {{winsecid John Doe}}

For more information about global.beg, see [Chapter 13](#), [“Including the Same Files and Commands in Many Documents”](#).

■ ■ ■

Index

Symbols

.mtd documents 57

Numerics

3270 emulation
 connection guidelines 19
 notification message channel 124
5250 connection guidelines 19

A

abort FCL code 189
action XML Interface function 143
activate InternetLink Module 177
addcopies FCL code 189
addressing documents 65
alternative fax number 65
altfax FCL code 189
approval FCL code 189
ASCII text files 12
ATTACH command 179, 181
attach FCL code 190
attachments
 cover sheets 82
 example document 106
 examples 179
 FCL codes for 105

file conversion 105
in native file format 179, 181
inline via HTTP/S for XML Interface 144
maximum number of pages 47
multipart MIME via HTTP/S for XML
 Interface 144
with IBM WebSphere MQ for the XML
 Interface 146

B

background forms
 creating 71
 example 71
 linking to a document 72
 overview 71
base 64 184
BASE64
 encoding a binary data file for the XML
 Interface 147
 for XML attachments with IBM WebSphere
 MQ 146
 for XML file attachments 146
batch FCL code 190
BEGIN EMAIL command 182
begin FCL code 190
BEGIN MIME command 182
beginvcvt FCL code 191

BEGINNATIVE command 179, 182
billing FCL code 191
binary data with XML Interface 147
bm FCL code 191
box FCL code 192
box with fill color, example in FCL
 document 89
box, example in FCL document 88, 90
broadcasting
 cover sheets in 69
 creating a list of recipients 67
 example 68
 with FCL codes 66
Bufdir.exe
 creating a directory scan 28
 testing directory scanning 35
 testing TCP/IP port connections 36
Buffer.exe 13

C

Capture.exe
 creating a serial capture 29
 testing serial port connections 36
capturing data via a named pipe 27
capturing data via a TCP/IP port for an IBM
 WebSphere MQ connection 32
CC command 184

- certified email, sending with FCL code 65
- cleartabs FCL code 192
- comment FCL code 192
- company name, default for cover sheet 46
- connection channels
 - creating 25-26
 - directory scanning 28
 - named pipe 27
 - serial port 29
 - TCP/IP port 31
 - TCP/IP port for IBM WebSphere MQ 32
 - testing directory scanning 35
 - testing printer 35
 - testing serial port 36
 - testing TCP/IP port 36
- connection guidelines
 - 3270 emulation 19
 - 5250 emulation 19
 - FTP 19
 - LPR 19
 - TCP/IP 19
- contact FCL code 193
- context-sensitive help 8
- convert XML to FCL 141
- cover FCL code 193
- cover sheets
 - attaching with FCL code 82
 - broadcasting with 78
 - creating 78
 - default 45, 78
 - default company name 46
 - default CSID 46
 - example 78
 - example FCL document 84
 - FCL codes for 81-82
 - for broadcasting 69
 - keywords 79-81
 - logic 83
 - with XML Interface 147

- covertext FCL code 194
- create InternetLink documents 178
- csi FCL code 194
- CSID, default for cover sheet 46
- current units, setting default for FCL 47
- customer support 9

D

- database notification channel 122
- database notifications 122
- date FCL code 195
- dbnotify.exe 123
- debug mode, Java API 174
- default
 - company name for cover sheet 46
 - cover sheet 45, 78
 - CSID for cover sheet 46
 - fonts 98
 - image quality 46
 - margins 94
 - page length 57
 - page orientation 101
 - printer 45
 - tabs 96
 - units of measure for FCL 47
- delay FCL code 195
- dept FCL code 195
- directory scanning 28
- document data flow overview 12
- document types, choosing 178
- documents
 - creating with FCL 40
 - specifying destination 65
- documents, FCL requirements for 178
- DTDs
 - XML_FAX_ACTION_REPLY.dtd 156

- XML_FAX_NOTIFICATION.dtd 156
- XML_FAX_QUERY_REPLY.dtd 154
- XML_FAX_SUBMIT_REPLY.dtd 153

E

- email FCL code 195
- email messages
 - document content for 178
 - send notification when a fax fails 181
- email server name 177
- email, sending with FCL code 65
- empid FCL code 196
- encoding
 - example using BASE64 to encode a binary data file for the XML Interface 147
 - for XML file attachments 146
 - for XML file attachments with IBM WebSphere MQ 146
- end FCL code 196
- endcovertext FCL code 196
- endcvr FCL code 196
- ENDNATIVE command 184
- endpoly FCL code 196
- Event Log 46
- examples
 - background form 71
 - box in FCL document 88, 90
 - box with fill color in FCL document 89
 - broadcasting 68
 - cover sheet with document 78, 84
 - document with attachment 106
 - document with signature 107
 - faxing from Java API 168
 - FCL documents 41

- fonts in documents 98
- HTTP/S post with XML Interface 144
- line in FCL document 89-90
- notification message 128
- notification message process 111
- notification message template 117
- page orientation 102
- setting margins for document 93
- tabs in documents 95
- XML Interface documents 157

executable files, list of 187

execute FCL code 196

F

failed fax 181

fax

- documents with FCL 65
- example in Java API 168
- maximum number of pages to attach 47
- maximum page length 47
- minimum page length 47
- reduce image size 47

fax FCL code 197

fax image resolution 102

fax image, indented in document 47

fax number

- alternative 65
- printing document if missing 57

FCL code

- abort 189
- addcopies 189
- adding graphics with 108
- adding signatures with 106
- altfax 189
- approval 189
- attach 190
- attaching files with 105
- background form for a document 72

- batch 190
- begin 190
- begincvt 191
- billing 191
- bm 191
- box 192
- broadcasting with 66
- cleartabs 192
- comment 192
- contact 193
- converting from XML 141
- cover 193
- covertext 194
- creating documents with 39
- creating notifications with 128
- creating shapes with 87-88
- csi 194
- date 195
- delay 195
- dept 195
- email 195
- empid 196
- end 196
- endcovertext 196
- endcvt 196
- endpoly 196
- example documents 41
- execute 196
- fax 197
- fax image quality 103
- ff 197
- file 197
- fillbox 197
- font 198
- fonts for documents 98
- for cover sheets 81-82
- for scheduling transmission 109-110
- form 199
- form feed 94

- in notification messages 129-130
- include 199
- including external files with 75
- libdoc 200
- line 200
- lineto 200
- linewidth 201
- list 201
- list of 189
- lm 201
- lookup 201
- lp 202
- lpr 202
- moveto 202
- no cover 202
- note 203
- notifyhost 203
- onerror 203-205
- onsuccess 205-207
- orient 207
- owner 207
- page orientation 101
- place 207
- placeall 207
- placelast 208
- placexy 208
- position 209
- preview 209
- printer 202
- printing documents with 70
- printnow 209
- priority 209
- quality 210
- rbox 210
- rem 210
- replyto 210
- rfillbox 211
- rline 211
- rlineto 211

- rmoveto 212
- rstartpoly 212
- rti 212
- sending documents to a file with 70
- sending documents with 65
- sending one document to many recipients with 66
- settab 213
- setting margins with 93
- setting tabs with 94
- sign 213
- signature 213
- sms 213
- smsmsg 213
- startpoly 214
- syntax 41
- tab 214
- termid 214
- time 215
- tm 215
- trandid 215
- type 216
- underline 217
- uniqueid 217
- units 217
- user1 218
- userid 218
- utc 218
- voice 218
- winsecid 219
- FCL documents
 - code format 41
 - creating 39, 40
 - fax when recipient data is missing 138-139
 - print when recipient data is missing 138-139

- required codes 40
- FCL, required in InternetLink documents 178
- ff FCL code 197
- field types, MapText 59-62
- fields, filter template 59-62
- file conversion of attachments 105
- file FCL code 197
- file transport command line in XML Interface 145
- file transport, with XML Interface 145
- fillbox FCL code 197
- filter mode 12
- filter template
 - creating 52-59
 - fields 59-62
 - input channel for documents 64
 - maintaining 51
 - planning 51
 - preview 58
 - troubleshooting 63-64
- font FCL code 98, 198
- fonts
 - default 98
 - example 98
 - FCL code for 98
 - overview 97
 - strikeout 98
 - TrueType 97
 - underline 98
 - Windows 97
- form FCL code 199
- form feed FCL code 94
- format
 - for FCL codes 41
- FROM command 184
- FTP connection guidelines 19

- FTP notification channel 125

G

- Generate Image function 58
- global include files
 - adding data to 75
 - linking to documents 75
 - names 74
 - overview 73
- graphic, adding to document with FCL code 108

H

- help, online 8
- host name 177
- HTTP XML Interface transport 143
- HTTP/S XML Interface transport 144

I

- IBM WebSphere MQ
 - attachments with the XML Interface 146
 - creating an input 32
 - notification channel 126
 - testing notification channel 131
 - XML Interface transport method 146
- ID, unique, in Java API 172
- image quality, setting default 46
- image size, reducing 47
- image type 178
- IMAGETYPE command 178, 185
- include FCL code 199
- include files 179
 - adding data to global 75
 - content of 73
 - creating 74
 - creating list of recipients for broadcasting 66

- global 73
- include FCL code 75
- linking to documents 75
- names of global 74
- overview 73
- standard 73
- indentation (pixels) 47
- inline 183
- inline attachments via HTTP/S for XML
 - Interface 144
- input channels
 - creating 25-26
 - for filter template documents 64
- installing
 - Java API 166
 - XML Interface 141
- Integration Module Configuration program,
 - starting 45
- Integration Module setup wizard, starting 21

J

- Java API
 - action on a document 172-174
 - debug mode 174
 - fax example 168
 - installing 166
 - querying a fax 170-171
 - sending documents 167
 - status codes 175

K

- keywords
 - for cover sheets 79-81
 - for notification message
 - templates 113-115

L

- landscape page orientation 101
- libdoc FCL code 200
- line FCL code 200
- line, example in FCL document 89-90
- lineto FCL code 200
- linewidth FCL code 201
- list FCL code 201
- lm FCL code 201
- lookup FCL code 201
- lookup tables 11
- Lotus Notes notification channel 127
- lp FCL code 202
- LPR connection guidelines 19
- lpr FCL code 202
- LPR, connecting with a named pipe 27

M

- Makedoc.exe 13
- MapText
 - .mtd documents 57
 - starting 52
- MapText field types 59-62
- margins
 - example document 93
 - FCL code 93
 - setting default 94
- maximum attached pages 47
- mediatype 184
- message body 178
- message body text in XML Interface 147
- Microsoft Exchange notification channel 127
- MIME attachments with XML Interface 144
- Mime.inc 179
- Mimesend.exe 13
- moveto FCL code 202
- Mqget.exe

- command line syntax 33
- creating an IBM WebSphere MQ
 - input 32
 - testing connections 36
- multipart MIME attachments 144

N

- named pipe, creating a capture 27
- native files, attaching to documents 179, 181
- native mode 12
- new page FCL code 94
- nocover FCL code 202
- note FCL code 203
- notification message channels
 - command line 120
 - creating 118
 - Notify.exe 120
 - overview 118
 - testing 131-134
 - testing IBM WebSphere MQ 131
- notification messages 181
 - creating communication channels 118
 - creating templates 112-113
 - example document 128
 - example of process flow 111
 - example template 117
 - FCL codes in 129-130
 - keywords for 113-115
 - overview 12, 111
 - retaining copies on RightFax server 119
- notification XML Interface function 143
- notifications
 - fax a copy of document upon
 - transmission 135-136
 - fax a copy of document upon transmission
 - failure 137-138
 - fax document when recipient data is
 - missing 138-139

- print a copy of document upon transmission [135-136](#)
- print a copy of document upon transmission failure [137-138](#)
- print document when recipient data is missing [138-139](#)
- sending notifications to a database [122](#)
- sending notifications via SMS [123](#)
- vs. submit-reply XML Interface function [143](#)
- Notify.exe [13, 120](#)
- notifyhost FCL code [203](#)
- Nplisten.exe
 - creating a named pipe capture [27](#)
 - testing printer connections [35](#)

O

- ONERROR command [181](#)
- onerror FCL code [203-205](#)
- online help [8](#)
- ONSUCCESS command [182](#)
- onsuccess FCL code [205-207](#)
- opening
 - Integration Module Configuration program [45](#)
 - Integration setup wizard [21](#)
 - MapText [52](#)
- orient FCL code [207](#)
- overview, document data flow [12](#)
- owner FCL code [207](#)

P

- page length
 - default [57](#)
 - maximum [47](#)
 - minimum [47](#)
- page orientation

- default [101](#)
- example [102](#)
- Parse.exe [13](#)
- PCL (print control language) files [12](#)
- PCX image type [178](#)
- PDF image type [178](#)
- place FCL code [207](#)
- placeall FCL code [207](#)
- placelast FCL code [208](#)
- placexy FCL code [208](#)
- Portlstn.exe [31](#)
- portrait page orientation [101](#)
- position FCL code [209](#)
- PostScript files [12](#)
- preview a filter template [58](#)
- preview FCL code [209](#)
- print control language (PCL) files [12](#)
- print copies if no fax number [57](#)
- printer FCL code [202](#)
- printer, setting default [45](#)
- printing with FCL [70](#)
- printnow FCL code [209](#)
- priority FCL code [209](#)
- programs, list of [187](#)

Q

- quality FCL code [210](#)
- query XML Interface function [143](#)
- querying a fax with Java API [170-171](#)
- quotedprintable [184](#)

R

- rbox FCL code [210](#)
- recipients
 - creating a list for broadcasting [67](#)
 - specifying addresses [65](#)
- rem FCL code [210](#)

- replyto FCL code [210](#)
- requirements, system [177](#)
- resolution, fax [102](#)
- rfillbox FCL code [211](#)
- rfsms.exe [123](#)
- RFWebCon.dll library in XML Interface [144](#)
- RightFax
 - customer support [9](#)
 - training [8](#)
- rline FCL code [211](#)
- rlineto FCL code [211](#)
- rmoveto FCL code [212](#)
- rstartpoly FCL code [212](#)
- rti FCL code [212](#)

S

- scheduling transmission [109](#)
- schemas
 - XML Interface schemas [147-157](#)
 - XML_FAX_ACTION_REPLY.dtd [156](#)
 - XML_FAX_ACTION_SCHEMA [155](#)
 - XML_FAX_QUERY_REPLY.dtd [154](#)
 - XML_FAX_QUERY_SCHEMA [153](#)
 - XML_FAX_SUBMIT_REPLY.dtd [153](#)
 - XML_FAX_SUBMIT_SCHEMA [148](#)
- send documents as email if faxes fail [181](#)
- sending
 - documents to a file with FCL [70](#)
 - one document to many recipients [66](#)
 - one document to many recipients with include files [66](#)
- settab FCL code [213](#)
- shape FCL codes [87-88](#)
- sign FCL code [213](#)
- signature FCL code [213](#)
- signature, adding to document with FCL code [106](#)
- sms FCL code [213](#)

- SMS notification channel 123
- SMS notifications 123
- smsmsg FCL code 213
- SMTP host name 177
- SMTP notification channel 121
- standard include files
 - content of 73
 - creating 74
 - linking to documents 75
 - overview 73
- starting
 - Integration Module Configuration program 45
 - Integration Module setup wizard 21
 - MapText 52
- startpoly FCL code 214
- status codes
 - Java API 175
 - XML Interface 175
- SUBJECT command 185
- submit XML Interface function 142
- support, customer 9
- syntax for FCL codes 41
- system requirements 177

T

- tab FCL code 214
- tabs
 - example 95
 - FCL code 94
 - setting default 96
- TCP/IP connection guidelines 19
- TCP/IP port capture 31
- templates
 - creating cover sheet 78
 - creating notification 112-113
- termid FCL code 214

- testing
 - directory scanning connections 35, 36
 - printer connections 35
 - TCP/IP port connections 36
- text files 12
- text for message body with XML
 - Interface 147
- TIF image type 178
- time FCL code 215
- tm FCL code 215
- TO command 185
- training 8
- trandid FCL code 215
- transmission
 - fax a copy of document upon failure 137-138
 - fax a copy of document upon success 135-136
 - fax document when recipient data is missing 138-139
 - print a copy of document upon failure 137-138
 - print a copy of document upon success 135-136
 - print document when recipient data is missing 138-139
 - scheduling 109
- transmission quality
 - setting default 46
 - setting with FCL 103
- transports
 - file attachments with XML Interface 146
 - XML Interface 141
- trim trailing white space 47
- troubleshooting

- filter templates 63-64
- notification message channels 131-134
- TrueType fonts 97
- TYPE command 182
- type FCL code 216

U

- underline FCL code 217
- unique ID
 - in Java API 172
 - in XML Interface 142
- uniqueid FCL code 217
- units FCL code 217
- units of measure for FCL 47
- user1 FCL code 218
- userid FCL code 218
- utc FCL code 218

V

- voice FCL code 218

W

- Windows Event Log 46
- Windows fonts 97
- winsecid FCL code 219

X

- XML
 - converting to FCL 141
 - multipart MIME attachments 144
- XML Interface
 - attachments via HTTP/S 144
 - attachments with IBM WebSphere MQ 146

- example documents 157
- example HTTP/S post 144
- file transport attachments 146
- file transport command line 145
- inline attachments via HTTP/S 144
- installing 141
- multipart MIME attachments 144
- notification vs. submit-reply function 143
- RFWebCon.dll library 144
- schemas 147-157
- status codes 175
- text for cover sheet 147
- text for message body 147
- transport options and functions 141, 165
- unique ID 142
- using XSLT scripts 142
 - with binary data 147
- XML Interface functions
 - action 143
 - notification 143
 - query 143
 - submit 142
- XML Interface transports
 - file 145
 - HTTP/S 144
 - IBM WebSphere MQ 146
- XML_FAX_ACTION_REPLY.dtd 156
- XML_FAX_ACTION_SCHEMA 155
- XML_FAX_NOTIFICATION.dtd 156
- XML_FAX_QUERY_REPLY.dtd 154
- XML_FAX_QUERY_SCHEMA 153
- XML_FAX_SUBMIT_REPLY.dtd 153
- XML_FAX_SUBMIT_SCHEMA 148
- XSLT scripts with the XML Interface 142