

OpenText™ Fax

Integration Module Administrator Guide

Describes how the OpenText Fax system and the Integration Module work with any document that is sent as a fax, email, or over the Internet. It is designed to support applications that produce outputs that are traditionally printed.

FXNET240400-AGI-EN-01

OpenText™ Fax

Integration Module Administrator Guide

FXNET240400-AGI-EN-01

Rev.: 2024-Oct-24

This documentation has been created for OpenText™ Fax CE 24.4.

It is also valid for subsequent software releases unless OpenText has made newer documentation available with the product, on an OpenText website, or by any other means.

Open Text Corporation

275 Frank Tompa Drive, Waterloo, Ontario, Canada, N2L 0A1

Tel: +1-519-888-7111

Toll Free Canada/USA: 1-800-499-6544 International: +800-4996-5440

Fax: +1-519-888-0677

Support: <https://support.opentext.com>

For more information, visit <https://www.opentext.com>

© 2024 Open Text

Patents may cover this product, see <https://www.opentext.com/patents>.

Disclaimer

No Warranties and Limitation of Liability

Every effort has been made to ensure the accuracy of the features and techniques presented in this publication. However, Open Text Corporation and its affiliates accept no responsibility and offer no warranty whether expressed or implied, for the accuracy of this publication.

Table of Contents

1	Introduction	13
1.1	Features of the Integration Module	14
1.1.1	Formatting documents	14
1.1.2	Scheduling documents	14
1.1.3	Attaching cover sheets and other attachments	15
1.1.4	Including data from a lookup table	15
1.1.5	Sending notification messages	15
1.2	Document recognition	15
1.2.1	ASCII text files	15
1.2.2	PostScript files	16
1.2.3	PCL files	16
1.3	Document distribution	16
2	Connecting the OpenText Fax Integration Module	19
2.1	Guidelines for common connection methods	20
2.1.1	Line printer remote connection	21
2.1.2	3270 emulation	21
2.1.3	TCP/IP socket connection	21
2.1.4	File transfer protocol (FTP) connection	22
2.1.5	5250 emulation	22
2.1.6	IBM WebSphere MQ client for Windows	22
2.2	Using the Integration Setup Wizard	22
2.2.1	Configure the Integration Module service	23
2.2.2	Define inputs for the host application	23
2.2.3	Set notification actions and messages	24
2.2.4	Format documents	24
2.2.5	Copy configuration settings to other nodes	24
3	Configuring the Integration Module to receive data	25
3.1	Creating an input device	25
3.2	Setting up a named pipe capture	26
3.3	Setting up directory scanning	27
3.4	Setting up a serial capture	29
3.5	Setting up a TCP/IP port capture	32
3.6	Setting up an IBM WebSphere MQ connection	33
4	Creating filter templates	37
4.1	Filter templates	37
4.2	Preparing to create filter templates	39
4.2.1	Assigning filter templates to documents	40
4.2.2	Maintaining the filter template	40

4.3	Creating a new filter template	41
4.3.1	Step 1: Overlay the sample data on a MapText document	42
4.3.2	Step 2: Map the document data to MapText fields	44
4.3.3	Step 3: Establish page length, orientation, and background forms	47
4.3.4	Step 4: Insert FCL	47
4.3.5	Step 5: Preview the filter	48
4.4	Field types	49
4.5	Troubleshooting	53
4.5.1	The background form is not aligned with data	54
4.5.2	Document data does not match the background form	54
4.5.3	Extra files are not included	55
4.5.4	Documents are not addressed or are not sent correctly	55
4.6	Creating an input channel for filter documents	55
5	Reusing FCL commands across many documents	57
5.1	Include files	57
5.1.1	The content of include files	58
5.1.2	The placement of global include files	58
5.2	Creating include files	59
5.2.1	Inserting content into global include files	59
5.2.2	Storing include files	60
5.3	Linking include files	60
5.3.1	Linking global include files	60
6	Creating and attaching cover sheets and other files	61
6.1	Cover sheets	61
6.2	Creating cover sheet templates	62
6.2.1	Creating a basic cover sheet template	63
6.2.2	Adding blocks of text to a cover sheet from the document data	63
6.2.3	Keywords for cover sheets	64
6.2.4	Word-wrapping cover sheet note text	67
6.2.5	FCL for cover sheets	67
6.3	Attaching a cover sheet to a document with FCL	69
6.4	Cover sheet logic	70
6.5	Attaching other types of files	71
6.5.1	Converting attached files	72
6.5.2	Storing attachments	72
7	Defining fax page appearance	75
7.1	Drawing shapes	75
7.1.1	Using the most common commands that create shapes	76
7.2	Embedding signatures with FCL	81
7.2.1	Storing signatures	81

7.2.2	Embedding a signature	81
7.3	Embedding graphics with FCL	82
7.4	Setting margins	83
7.4.1	Setting margins for individual documents	83
7.4.2	Setting default margins for all documents	84
7.4.3	Starting a new page with the same margins as the previous page	84
7.5	Setting tabs with FCL	84
7.5.1	Setting tabs for individual documents	85
7.5.2	Setting default tabs for all documents	86
7.6	Selecting and configuring fonts	86
7.6.1	TrueType fonts	86
7.6.2	Support for other fonts	87
7.6.3	Selecting fonts	87
7.6.3.1	Selecting fonts for individual documents	87
7.6.3.2	Selecting a default font for all documents	88
7.6.4	Changing the appearance of fonts	88
7.7	Setting page orientation	89
7.7.1	Setting page orientation for individual documents	90
7.7.2	Setting default orientation for all documents	90
8	Document transmission and notifications	91
8.1	Scheduling document transmission	91
9	Creating notification messages with FCL	93
10	Creating notification templates	95
10.1	Template file keywords	96
10.2	The ^completiontime^ and ^transtime^ keywords	99
10.3	The ^faxstatus^ and ^statustype^ keywords	101
10.4	The ^retcode^ and ^statusstring^ keywords	102
10.5	Sample notification templates	104
11	Creating notification channels	107
11.1	Setting the number of retries for notifications	108
11.2	Creating notification channels in a Shared Services environment	109
11.3	Methods used for notification messages	109
11.4	Sending notifications via SMTP	111
11.5	Sending notifications to a database	113
11.5.1	Verifying that the database notification was sent	114
11.6	Sending notifications to an SMS-enabled device	114
11.7	Sending notifications via 3270 emulation	116
11.8	Sending notifications via FTP	117
11.9	Sending notifications via IBM WebSphere MQ	118

11.10	Sending notifications to Notes	120
11.11	Sending notifications to Microsoft Exchange	121
12	Including the {{Notifyhost}} command in documents	123
12.1	Creating the {{Notifyhost}} command	123
12.2	Other FCL used in notifications	124
13	Testing that the host application is correctly receiving notifications	127
13.1	Testing IBM WebSphere MQ connections	128
14	Troubleshooting notification messages	131
15	Setting up actions on document transmission	135
15.1	Sending notifications of successful and failed transmissions	135
15.1.1	Setting the default success and error actions	135
15.1.2	Overriding the default success or error action	136
15.2	Performing actions on documents with FCL errors	137
16	Programming for the OpenText Fax XML Interface	139
16.1	Introduction to the OpenText Fax XML Interface	139
16.2	Installing the XML Interface	140
16.2.1	Minimum system requirements	140
16.2.2	Installing the XML Interface	140
16.3	XML Interface functions	141
16.3.1	Submit	141
16.3.2	Query	142
16.3.3	Action	142
16.3.4	Notification	142
16.4	Transports	143
16.4.1	HTTP or HTTPS transport	143
16.4.2	Example HTTP or HTTPS post	144
16.4.3	HTTP or HTTPS attachments	144
16.4.4	HTTP or HTTPS notifications	145
16.4.5	File transport	145
16.4.6	File transport attachments	146
16.4.7	IBM WebSphere MQ transport	146
16.4.8	IBM WebSphere MQ attachments	147
16.5	Body and cover text	147
16.5.1	Using plain text data	147
16.5.2	Using formatted text data	148
16.5.3	Using binary data	148
16.6	The schemas	149
16.6.1	XML_FAX_SUBMIT_schema	149

16.6.2	XML_FAX_SUBMIT_REPLY.dtd	152
16.6.3	XML_FAX_QUERY_schema	153
16.6.4	XML_FAX_QUERY_REPLY.dtd	153
16.6.5	XML_FAX_ACTION_schema	154
16.6.6	XML_FAX_ACTION_REPLY.dtd	155
16.6.7	XML_FAX_NOTIFICATION.dtd	155
16.7	Sample documents based on the schemas	156
16.7.1	XML_FAX_SUBMIT	156
16.7.2	XML_FAX_SUBMIT_REPLY	158
16.7.3	XML_FAX_QUERY	158
16.7.4	XML_FAX_QUERY_REPLY	158
16.7.5	XML_FAX_ACTION	159
16.7.6	XML_FAX_ACTION_REPLY	159
16.7.7	XML_FAX_NOTIFICATION	159
16.7.8	Testing the sample files	160
17	Creating FCL documents	161
17.1	The format of FCL commands	161
17.2	Setting defaults for FCL documents	162
17.2.1	Opening the Integration Configuration Program	162
17.2.2	Setting general defaults	162
17.2.3	Setting defaults for FCL processing	163
17.3	Sending basic FCL jobs	164
17.3.1	Sending one document to a single fax number	164
17.3.2	Sending one document to many recipients	164
17.3.3	Broadcast faxing with an include file	165
17.3.4	Creating a broadcast list	167
17.3.5	Using cover sheets in a broadcast	167
17.3.6	Sending documents to a printer or file with FCL	169
17.3.7	FCL Dashboard	170
17.4	Creating and linking background forms	170
17.4.1	Background forms	170
17.4.2	Creating background forms	171
17.4.3	Linking background forms	171
17.5	Example FCL documents	171
18	Programming for the OpenText Fax API for Java	175
18.1	Installing the API for Java	176
18.1.1	Minimum system requirements	176
18.1.2	Installing the Java Interface	176
18.1.3	Configuring the Java development computer to use the OpenText Fax Java interface	177
18.2	Sending outbound documents	177

18.2.1	Sample code for creating an outbound document	178
18.3	Querying documents	179
18.3.1	Sample code for querying a document	180
18.4	Performing actions on documents	181
18.4.1	Sample code for performing an action on a document	182
18.5	Using debug mode	183
18.6	Error and status codes for XML- and Java-based documents	183
19	Creating FCL documents with InternetLink commands	187
19.1	System requirements	187
19.2	Verifying the name of the email server	187
19.3	InternetLink FCL commands	188
19.3.1	Choosing document types	189
19.3.2	Choosing image types	189
19.3.3	Using include files with the InternetLink Module	189
19.4	Attaching native files to InternetLink documents	189
19.4.1	Example FCL for InternetLink documents that have attachments	190
19.5	Sending documents as email if faxes fail	192
19.6	Receiving notification when a fax fails and is sent as an email	193
20	OpenText OpenText Fax Connector for Oracle Purchasing 11i	195
20.1	Activating the Oracle Purchasing 11i Connector	195
20.2	How the Oracle Purchasing 11i Connector works	195
20.2.1	Sending documents from Oracle 11i	195
20.2.2	Document templates	195
20.3	Editing the default templates	196
20.3.1	Adding a graphic image or a background form	197
20.3.2	Adding a cover sheet	197
20.3.3	Configuring the layout	198
20.3.4	Faxing a copy of a document that was transmitted	198
20.3.5	Faxing a copy of a document that cannot be transmitted	199
20.3.6	Specifying the notification method	199
20.3.7	Adding FCL code to the template	200
20.4	Sending documents by email	200
21	OpenText Fax Integration Module programs	201
22	FCL Commands	203
22.1	ABORT	203
22.2	ADDCOPIES	203
22.3	ALTFAX	203
22.4	APPROVAL	204
22.5	ATTACH	204

22.6	BATCH	205
22.7	BEGIN	205
22.8	BEGINCVT	206
22.9	BEGINNATIVE	206
22.10	BILLING	211
22.11	BILLING2	211
22.12	BM	211
22.13	BOX	212
22.14	CC	212
22.15	CLEAR TABS	213
22.16	COMMENT	213
22.17	COMMENTFORRECORDS	213
22.18	COMPANY	214
22.19	CONTACT	214
22.20	CONVERSION_BIAS	214
22.21	COVER	215
22.21.1	Specific cover names	215
22.22	COVERTEXT	216
22.23	CSI	216
22.24	DATE	217
22.25	DELAY	217
22.26	DEPT	218
22.27	EMAIL	218
22.28	EMPID	218
22.29	END	218
22.30	ENDCOVERTEXT	219
22.31	ENDCVT	219
22.32	ENDNATIVE	219
22.33	ENDPOLY	219
22.34	EXECUTE	220
22.35	FAX	220
22.36	FF	221
22.37	FILE	221
22.38	FILLBOX	222
22.39	FONT	222
22.40	FORM	224
22.41	FROM	225
22.42	IMAGETYPE	225
22.43	INCLUDE	225
22.44	LIBDOC	226
22.45	LINE	226
22.46	LINETO	227

22.47	LINEWIDTH	227
22.48	LIST	228
22.49	LM	228
22.50	LOOKUP	229
22.51	LP, LPR, or PRINTER	229
22.52	MOVETO	230
22.52.1	Using the {{moveto}} command in a Unix system	230
22.53	NOCOVER or NO_COVER	231
22.54	NOTE	231
22.55	NOTIFYHOST	231
22.56	ONERROR	232
22.57	ONSUCCESS	234
22.58	ORIENT	236
22.59	OWNER	236
22.60	PDFTYPE	236
22.61	PDFOWNER	237
22.62	PDFUSER	237
22.63	PLACE	238
22.64	PLACEALL	239
22.65	PLACELAST	239
22.66	PLACEXY	240
22.67	POSITION	240
22.68	PREVIEW	241
22.69	PRINTER	241
22.70	PRINTNOW	241
22.71	PRIORITY	241
22.72	PRIVATEFAX	242
22.73	RBOX	242
22.74	REM	243
22.75	REPLYTO or REPLY_TO	243
22.76	RETRYCOUNT	243
22.77	RETRYINTERVAL	244
22.78	RFILLBOX	244
22.79	RLINE	245
22.80	RLINETO	245
22.81	RMOVETO	246
22.82	RSTARTPOLY	246
22.83	RTI	247
22.84	SETTAB	247
22.85	SIGN, SIGNED, or @	248
22.86	SIGNATURE	248
22.87	SMS	248

22.88	SMSMSG	249
22.89	STARTPOLY	249
22.90	SUBJECT	250
22.91	TAB	250
22.92	TERMID	251
22.93	TIME	251
22.94	TM	251
22.95	TO	252
22.96	TRANID	252
22.97	TYPE	252
22.98	UNDERLINE	255
22.99	UNIQUEID or UNIQUE_ID	255
22.100	UNITS	256
22.101	USER1, USER2, or USER3	256
22.102	USERID	257
22.103	UTC	257
22.104	VOICE	258
22.105	WINSECID	258
22.105.1	Controlling the creation of users for faxes sent from non-existent users	259

Chapter 1

Introduction

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

The 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 OpenText Fax system can integrate with many applications. The integration options are listed in the following table, along with the OpenText Fax documentation where you can find more information.

Integration option	For more information
<i>Integration Module</i> services on the OpenText Fax server with Facsimile Command Language (FCL)	Chapters 2-9 in this guide, and the <i>OpenText Fax - Administrator Guide (FXNET240400-AGD)</i>
OpenText Fax XML Interface	“Programming for the OpenText Fax XML Interface” on page 139
OpenText Fax API for Java	“Programming for the OpenText Fax API for Java” on page 175
OpenText Fax C, C++, and Visual Basic API	www.opentext.com

To meet the needs of small, medium, and large enterprises, the Integration Module comes in two versions: the Integration Module which includes full functionality, and the OpenText Fax Business Integration Module which includes a limited set of features for use by smaller organizations.

Integration Module

- 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.

OpenText Fax Business Integration Module

- 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.

The instructions in this document apply to both versions of the module unless otherwise specified.

1.1 Features of the Integration Module

The Integration Module provides the following features.

1.1.1 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.

1.1.2 Scheduling documents

In addition to the scheduling control provided by OpenText Fax, 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.

1.1.3 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.

1.1.4 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.

1.1.5 Sending notification messages

As a document is sent via OpenText Fax, 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 emailed 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 email address.
- Document data such as the number of pages, transmission date and time, 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.

1.2 Document recognition

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

1.2.1 ASCII text files

To process text files, 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. It also contains the FCL that is required to create and send the document. The FCL is “laid over” the document according to the map. This is known as **filter mode**.

1.2.2 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.

1.2.3 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.

1.3 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.

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 “[OpenText Fax Integration Module programs](#)” on page 201.

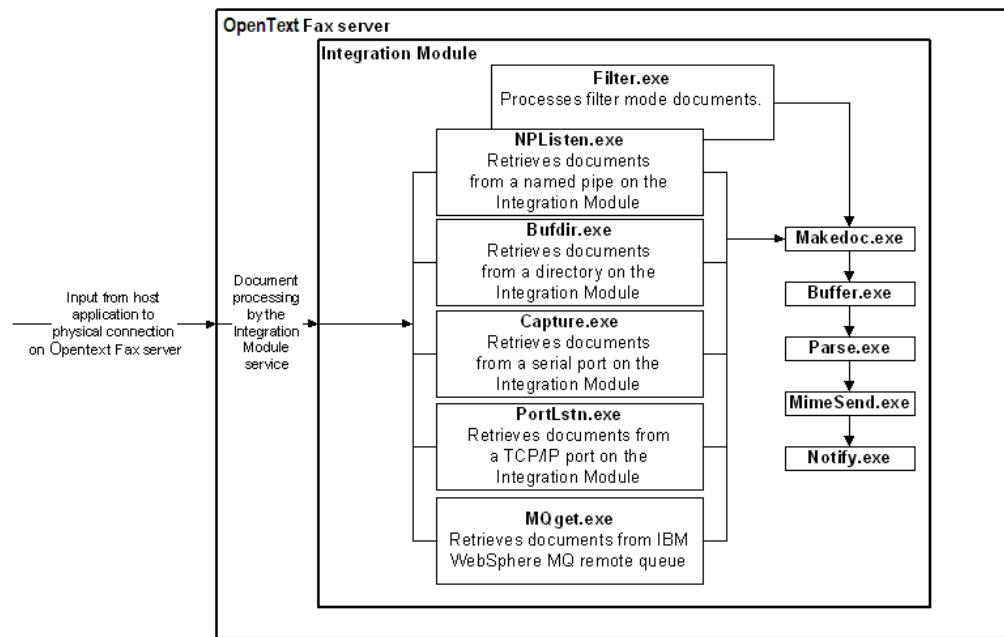


Figure 1-1: Flow of documents in the Integration Module

Figure 1-2 shows how FCL documents are processed after the Integration Module receives the data stream from a host application.

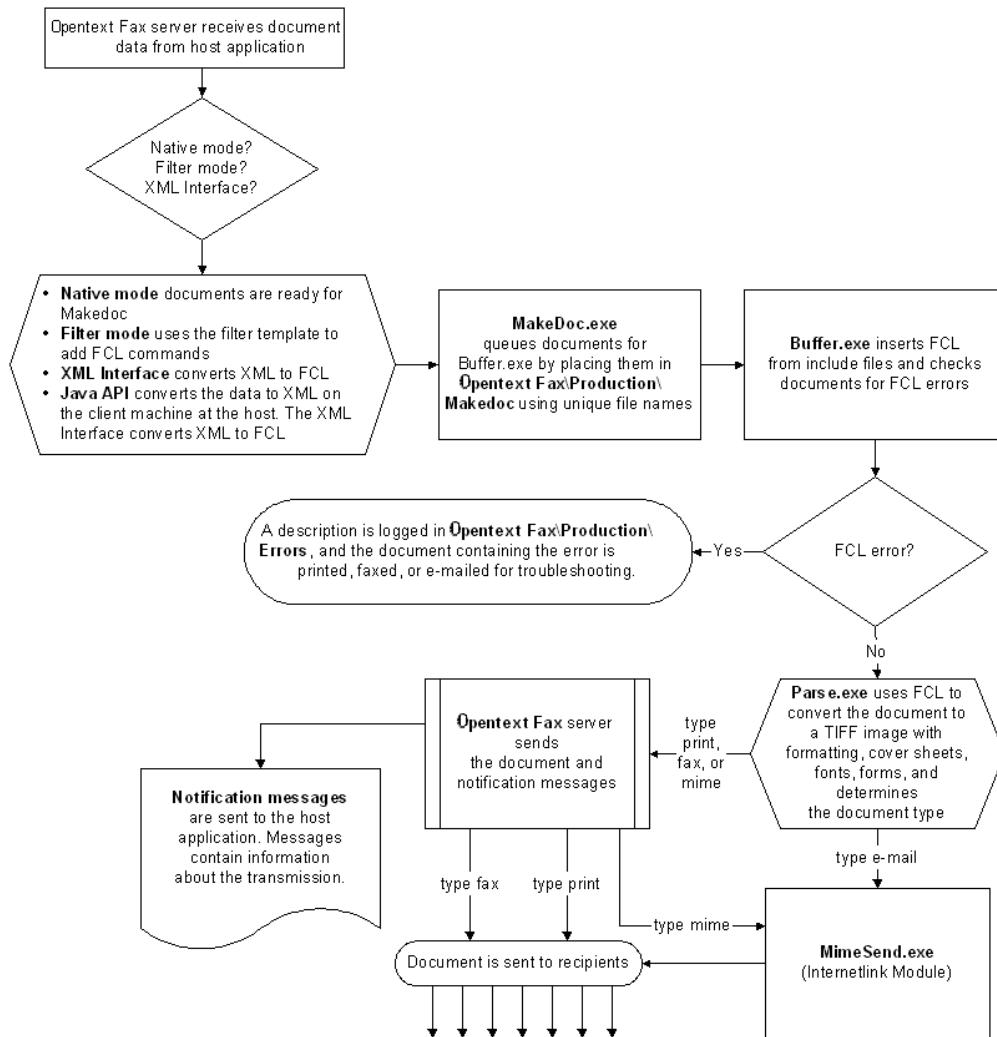


Figure 1-2: OpenText Fax Integration Module data flow

Chapter 2

Connecting the OpenText Fax Integration Module

The Integration Module software (both standard and business versions) is installed automatically during the OpenText Fax installation. To enable the functionality of these modules, you must license and then enable them on the server.

For information on enabling or adding new components to an OpenText Fax installation, see *OpenText Fax - Administrator Guide (FXNET240400-AGD)*.

After the Integration Module is enabled, you must connect the module to a host computer. This section explains the most common methods for connecting a host computer to the Integration Module computer for FCL-based documents.

The following figures illustrate the most common connection methods.

Some connections require third-party software that OpenText does not provide or support.

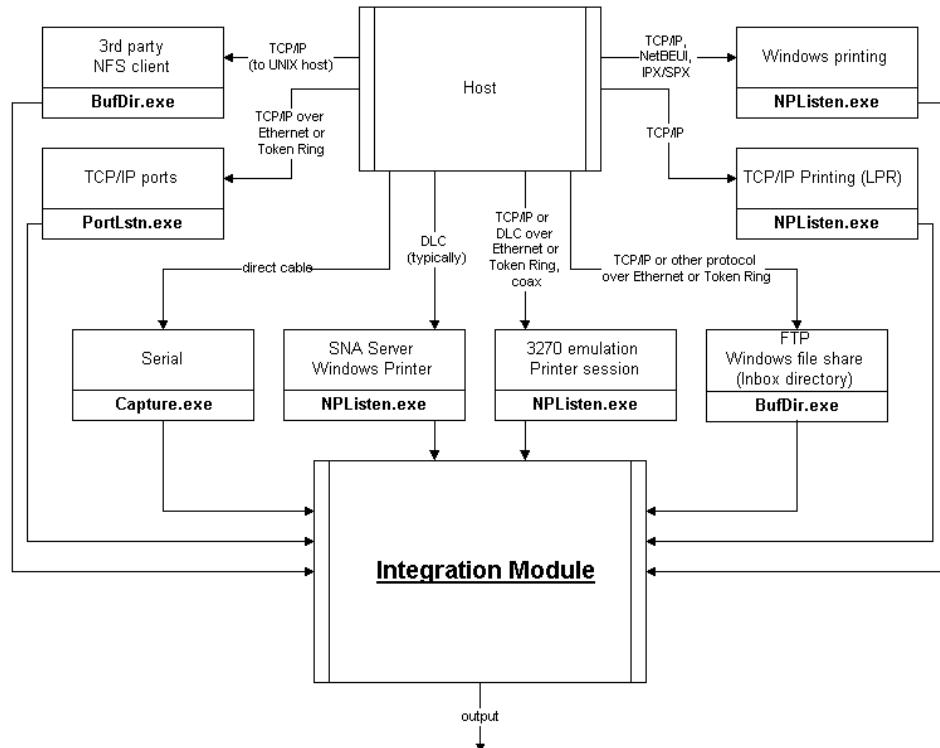


Figure 2-1: Connections to applications that generate FCL or ASCII data

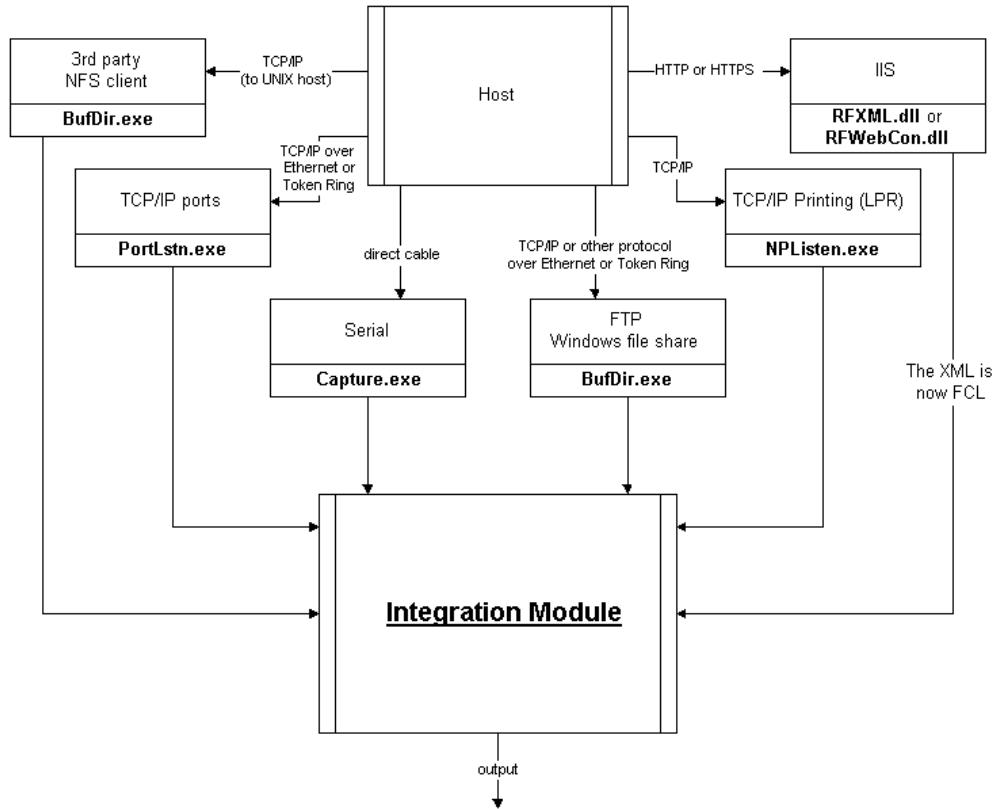


Figure 2-2: Connections to host applications that generate XML and Java data

2.1 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 in “[Line printer remote connection](#)” on page 21.
- 3270 emulation in [3270 emulation](#).
- TCP/IP socket connection in “[TCP/IP socket connection](#)” on page 21.
- File Transfer Protocol (FTP) connection in “[File transfer protocol \(FTP\) connection](#)” on page 22.
- 5250 emulation in [5250 emulation](#).
- IBM® WebSphere MQ® client versions 7, 7.5, 8, 9, 9.1.3.0, and 9.2.10 for Microsoft Windows in “[IBM WebSphere MQ client for Windows](#)” on page 22.

After the communication methods are established, you must configure the Integration Module to receive data via those methods. See “[Configuring the Integration Module to receive data](#)” on page 25.

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

2.1.1 Line printer remote connection

To use a line printer remote connection, set up the host system to print to the OpenText Fax 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 OpenText Fax Server. See “[Setting up a named pipe capture](#)” on page 26.

2.1.2 3270 emulation

You can set up a direct connection from the OpenText Fax 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 a better choice and is easier to configure if the mainframe has a TCP/IP connection.

When 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 supports 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. See “[Setting up a named pipe capture](#)” on page 26.

2.1.3 TCP/IP socket connection

You can set up a direct socket connection via TCP/IP to and from the OpenText Fax Server through any available port. This usually requires that you acquire or create connection software for the host system. See “[Setting up a TCP/IP port capture](#)” on page 32.

2.1.4 File transfer protocol (FTP) connection

The 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 OpenText Fax Server computer. See ["Setting up directory scanning" on page 27](#).

2.1.5 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 OpenText Fax Server via a serial cable connected between the converter and one of the COM ports on the OpenText Fax 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. ["Setting up a serial capture" on page 29](#).

2.1.6 IBM WebSphere MQ client for Windows

The Integration Module can communicate with an IBM WebSphere MQ channel via TCP/IP. See ["Setting up an IBM WebSphere MQ connection" on page 33](#).

2.2 Using the Integration Setup Wizard

The 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 it.
- 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.
- Copy the saved configuration settings to other nodes (available only on servers that are part of a Shared Services environment). See ["Copy configuration settings to other nodes" on page 24](#).

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. Review the topics in this guide before you begin using the wizard.

To start the wizard

1. On the **Start** menu, click **OpenText Fax**, and then click **OpenText Fax 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 **OpenText Fax Integration Module**.
-  **Note:** If you do not see Integration Module in the list, go to the Control Panel and double-click the OpenText Fax Integration Module from there.
4. The **Integration Module Configuration** window opens.
 5. Click **Integration Setup Wizard**. The wizard starts, and the **Integration Setup Wizard** window appears.

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

2.2.1 Configure the Integration Module service

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.
- Page length, page size, and reducing the image size to fit the page. See “[Setting defaults for FCL documents](#)” on page 162.

2.2.2 Define inputs for the host application

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, specify the communications protocol and transmission method, port, directory, or queue to monitor, and the configurations needed for each protocol. This includes naming the input, specifying the input type, and configuring the input. These settings are also described in “[Configuring the Integration Module to receive data](#)” on page 25.

2.2.3 Set notification actions and messages

You can define notification actions and messages.

You can set notification actions to:

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

You can set notification messages to:

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

Notification messages can provide descriptive information, such as whether 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 “[Document transmission and notifications](#)” on page 91.

2.2.4 Format documents

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

2.2.5 Copy configuration settings to other nodes

In this series of steps, you can copy the Integration Module configuration settings to servers that are part of a Shared Services environment.

To copy service configurations to other nodes

1. In the **OpenText Fax Integration Module**, complete the configuration settings, and then click **Apply** to save the changes.
2. Click **Copy to Other Nodes**.
The other nodes in your Shared Services environment appear in the **Copy the current settings to these shared services nodes** list.
3. Clear the check boxes of those nodes to which these settings should not be copied, and then click **OK**.

Chapter 3

Configuring the Integration Module to receive data

This section describes configuring the Integration Module to receive data. Create an input device for receiving or retrieving data by writing a command line for each type of input needed in the system.

 **Note:** The command lines are case-sensitive.

3.1 Creating an input device

To create an input device

1. On the **Start** menu, click **OpenText Fax**, and then click **OpenText Fax Enterprise Fax Manager**. The **Enterprise Fax Manager** window opens.
2. In the **Fax Servers** list, click the name of the server where the OpenText Fax Integration Module is running.
3. In the **Service Name** list, double-click **OpenText Fax Integration Module**.
4. In the left pane of the **Integration Module Configuration** window, click **Inputs**. The input settings appear.
5. Right-click **Inputs**, and select **Add Input Device** from the shortcut menu. The **Add Input Device** dialog box appears.
6. 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.
7. Enter the settings for the type of data input, as described in the following sections:
 - “Setting up a named pipe capture” on page 26
 - “Setting up directory scanning” on page 27
 - “Setting up a serial capture” on page 29
 - “Setting up a TCP/IP port capture” on page 32
 - “Setting up an IBM WebSphere MQ connection” on page 33
8. To create the settings for an input, 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.

3.2 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 `Nplistens.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.
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 `Nplistens.exe` command line options.

Setting	Command line option	Description
Named Pipe to Create	<code>-p pipename</code>	The name of the named pipe that <code>Nplistens.exe</code> will monitor for files. You can enter any name for the pipe in one of the following formats: pipename <code>\.\.\pipe\pipename</code>
Execute on Input (-c option)	<code>-c "makedoc \$\$"</code>	The command to execute on files received via the named pipe. <code>Makedoc.exe</code> begins the process of converting data from the host application into a fax. The variable <code>\$\$</code> indicates that all files should be processed with <code>Makedoc.exe</code> .

`nplistens.exe -c "makedoc $$" -p hplpr`

In this example, `Nplistens.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.

To test a named pipe connection

1. Make sure you created a printer on the OpenText Fax server for the named pipe input. This may be the HostFax printer.
2. Stop the Integration Module service.

3. Pause printing to the printer that was created for the named pipe input.
4. Send a test document from the host application to the printer. Typically this is done by copying the file from a command prompt.
5. Look in the print queue to verify that the test document has been sent to print.
6. Take the printer out of pause mode.
7. Open a command prompt and go to ..\OpenText Fax\Production\bin.
8. Go to the Integration Module Configuration window in Enterprise Fax Manager, and under Inputs highlight the pipe you're testing; copy the Complete Command Line text and paste it into the command prompt. Press **Enter**.
9. Now go to ..\OpenText Fax\Production\MakeDoc.
10. Open and examine the test document.
 - Documents that are created with filter templates (filter mode) are plain ASCII text.
 - Documents that are created in native mode are FCL documents-a combination of FCL commands and document data from the host application.
 - PCL or PostScript documents are in PCL or PostScript format with FCL commands on the first "false" page.
11. At the command prompt, stop the running application by pressing **Ctrl/C**.
12. Type "Buffer" or "Buffer.exe" and press **Enter**. The file in MakeDoc disappears, and a new file appears in ..\OpenText Fax\Production\Seq. 1.
13. At the command prompt, stop the running application by pressing **Ctrl + C**.
14. Type "parse" or "parse.exe" and press **Enter**. Parse completes the conversion of the file. The fax will appear in FaxUtil.
15. Re-start the Integration Module service.

3.3 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 processing. Having discrete subdirectories prevents duplication of faxes.

To set up directory scanning

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

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

Setting	Command line option	Description
Don't check for file in use	-u	This setting determines that Bufdir.exe does 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.

```
bufdir -c "makedoc $$" c:\program files\OpenText Fax\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.

To test the directory scanning connection

1. Stop Bufdir.exe by stopping the OpenText Fax 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 OpenText Fax Integration Module to verify that the test document was received.
4. Start the OpenText Fax Integration Module service in Enterprise Fax Manager.

3.4 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.
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 3-1: Add input device settings for a serial capture

Input setting	Command line option	Description
End Sequence	-S sequence	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 baud rate	The baud rate at which Capture .exe will scan the serial port.
Bits/Character	-C size	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) are converted to <CR><LF> (carriage return and line feed).
Left Brace Char	-L character	Specify one or more characters that form the left (opening) delimiter in documents that are received via this serial port.
Right Brace Char	-R character	Specify one or more characters that form the right (closing) delimiter in documents that are received via this serial port.
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 of each document that is received via this serial port.

Input setting	Command line option	Description
Stop Bits	-s stopbits	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.
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.

```
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.

To test a serial 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, go to OpenText Fax\Production\Makedoc and verify that the test document was received.
5. In Enterprise Fax Manager, start Buffer.exe by stopping and starting the OpenText Fax Integration Module service.

3.5 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.
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 3-2: Add input device settings for a TCP/IP port capture

Input setting	Command line option	Description
TCP/IP Port #	-p number	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.

```
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.

To test a TCP/IP port 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 Portlstn.exe command line.

4. In Windows Explorer, navigate to OpenText Fax\Production\Makedoc and verify that the test document was received.
5. In Enterprise Fax Manager, start Buffer.exe by stopping and starting the OpenText Fax Integration Module service.

3.6 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` uses 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.
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 <queue manager> -Q <queue> [<options>]
```

Table 3-3: Mqget.exe Command Line Options

Option	Description
<code>-C <channel></code>	Name to use for this connection channel.
<code>-H <hostname></code>	Fully qualified domain name of the IBM WebSphere MQ queue manager.
<code>-M <queue manager></code>	Queue manager for OpenText Fax that is defined on the IBM WebSphere MQ server.
<code>-Q <queue></code>	Queue to retrieve messages from. The OpenText Fax Integration Module will monitor and retrieve messages from this queue.
<code>-T</code>	Enable TLS.  Note: You must set the Queue Channel SSL Authentication to Optional .

Option	Description
<code>-S "<SSL CipherSpecs>"</code>	Specifies the SSL CipherSpecs of the queue channel SSL settings. For example: ANY_TLS12_OR_HIGHER Note: You must set the Queue Channel SSL Authentication to Optional .
<code>-R "<SSL key repository>"</code>	Specifies the SSL key repository of the queue manager SSL configuration settings. For example: C:\ProgramData\IBM\MQ\qmgrs\QM_APPLE\ssl\key\key Note: You must set the Queue Channel SSL Authentication to Optional .
<code>-c "makedoc \$\$"</code>	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.
<code>-d</code>	Display debugging output. This is helpful if you experience difficulty connecting to the server.
<code>-p<port></code>	TCP/IP port number to use for remote connection. The default is 1414.
<code>-1</code>	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.
<code>-t<CCSID></code>	Specifies the codeset name for a language. A list of the codeset IDs (CCSIDs) supported by WebSphere MQ is available from IBM.

```
mqget -C RF_Chан -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_Chан` 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`.

To set the MQGet polling interval

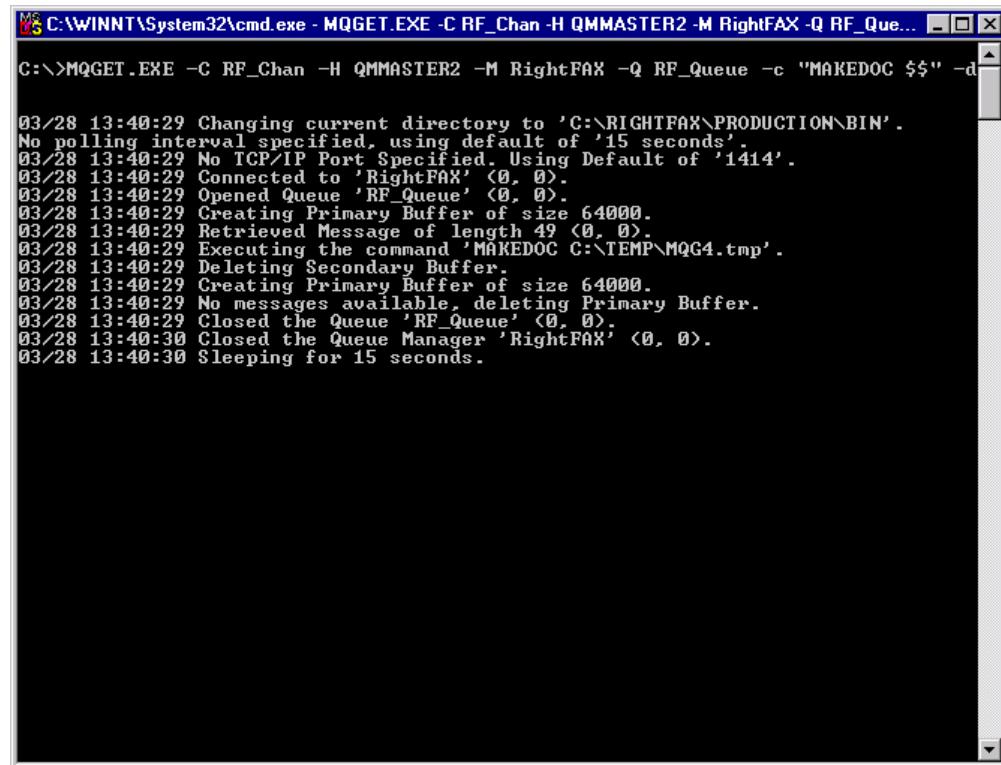
1. Log on to the OpenText Fax server as an administrator.

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 the Windows registry editor.

To test the IBM WebSphere 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, text similar to that shown in the following example appears. If the input connection is not successful, error messages appear instead.



```
MS 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'.
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 4

Creating filter templates

4.1 Filter templates

The 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 either native mode or filter mode.

Native 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.

Filter mode

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 section discusses the creation of filter templates that support filter mode.

Filter templates

Filter templates provide the following features:

- Data mapping provides sending information to the Integration Module, such as the recipient name, fax number, and email 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 email address.

Combined with the 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 OpenText Fax Server 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.

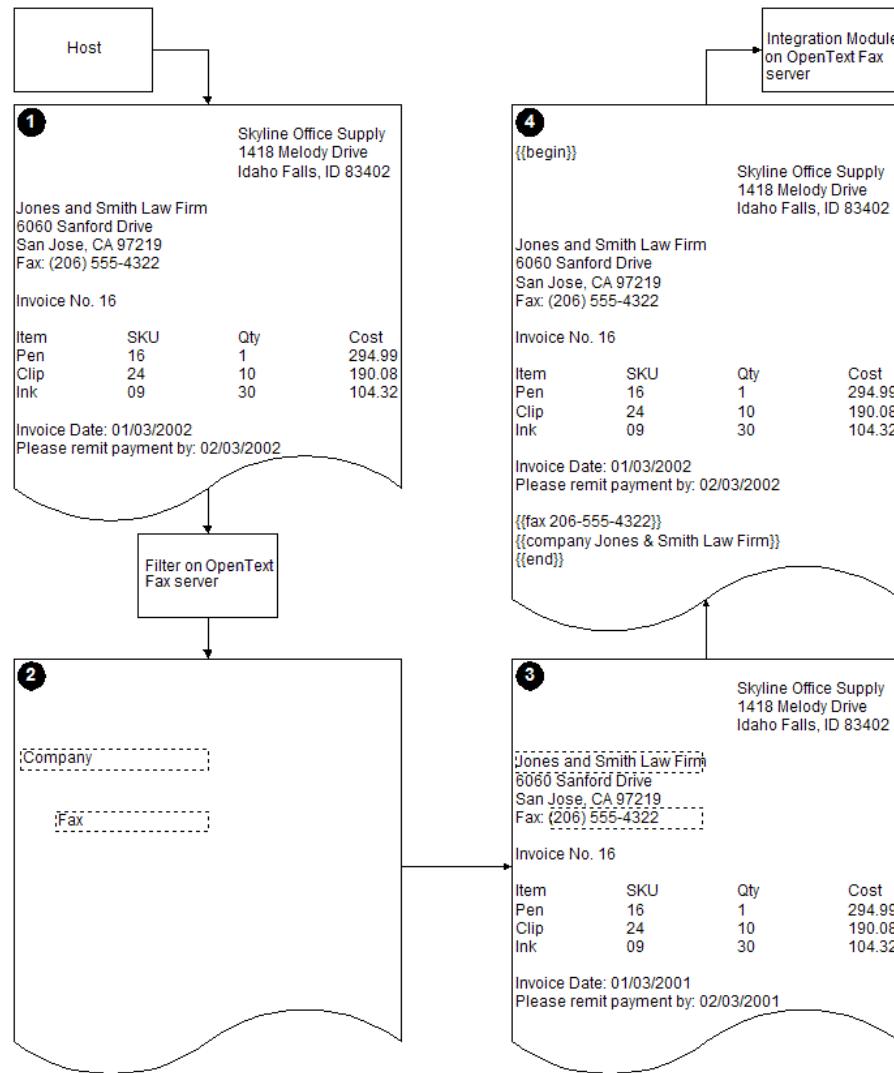


Figure 4-1: Filter mode document flow

4.2 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 with the data to 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 `OpenText Fax\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, then create the forms and save them in the appropriate folder on the OpenText Fax server. See “[Creating and linking background forms](#)” on page 170.
- Identify each cover sheet that should be included with each document, then create the cover sheets and save them in the appropriate folder on the OpenText Fax server. See “[Creating and attaching cover sheets and other files](#)” on page 61.
- Identify each graphic element, such as logos or signatures, to include in the filter template, then create the graphics and save them in the appropriate folder on the OpenText Fax server. See “[Defining fax page appearance](#)” on page 75.
- Create include files, if needed. See “[Reusing FCL commands across many documents](#)” on page 57.
- Create lookup tables, if needed using the {{lookup}} FCL command. See “[FCL Commands](#)” on page 203.
- Create TCP/IP print queues in the Integration Module . This is typically one print queue per filter.

4.2.1 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 “[Field types](#)” on page 49.
- Create up to 70 unique input channels (for the Integration Module) or 2 input channels (for the OpenText Fax Integration Business Module) 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 55.

4.2.2 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.

4.3 Creating a new filter template

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, refer to the instructions on the following pages:

- “Step 1: Overlay the sample data on a MapText document” on page 42.
- “Step 2: Map the document data to MapText fields” on page 44.
- “Step 3: Establish page length, orientation, and background forms” on page 47.
- “Step 4: Insert FCL” on page 47.
- “Step 5: Preview the filter” on page 48.

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

Table 4-1: Sample files installed with filter for production

Sample file	Location	Description
SampleData.txt	OpenText Fax\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	OpenText Fax\Production\Forms	An example of a form onto which you might print a purchase order. MapText merges SampleData.txt with SampleForm.tif to create an image of the document printed on a pre-printed form.

4.3.1 Step 1: Overlay the sample data on a MapText document

To overlay the sample data

1. On the OpenText Fax server in Windows Explorer, go to OpenText Fax\Production\Bin and run MapText.exe. The MapText window opens with a blank document called MapText1.
2. On the **Tools** menu, click **Overlay Data File**.
3. Select your sample document or SampleData.txt, and then click **Open**.

The horizontal and vertical rows of numbers at the top and to the left of the page illustrate that MapText filter templates are grids on which you create fields at x- and y-coordinates.

SampleData.txt shows Metropolitan Inc. purchase orders going to Universal Suppliers. The bill to address and contact information appear first, followed by the ship to address and contact information, and then Universal Suppliers as the recipient of the purchase order.

4.3.2 Step 2: Map the document data to MapText fields

In this step, you identify pieces of data in the document that the OpenText Fax server will use to process and send the documents. This includes:

- Locating the data on the page (creating a field for the data).
- Identifying the type of data in the field.
- Mapping facsimile command language (FCL) to the field.

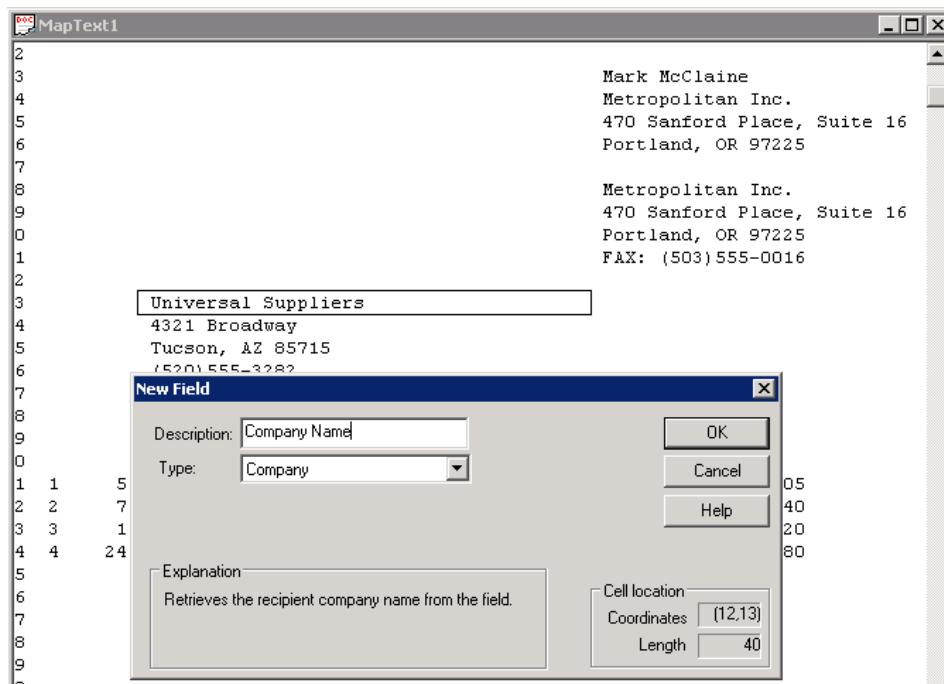
You can map host data to 25 FCL codes. The following steps map three host data elements (company name, fax number, and purchase order number) to three fields (company, fax, and comment).

For a detailed description of each type of field, see “[Field types](#)” on page 49.

To map the document data

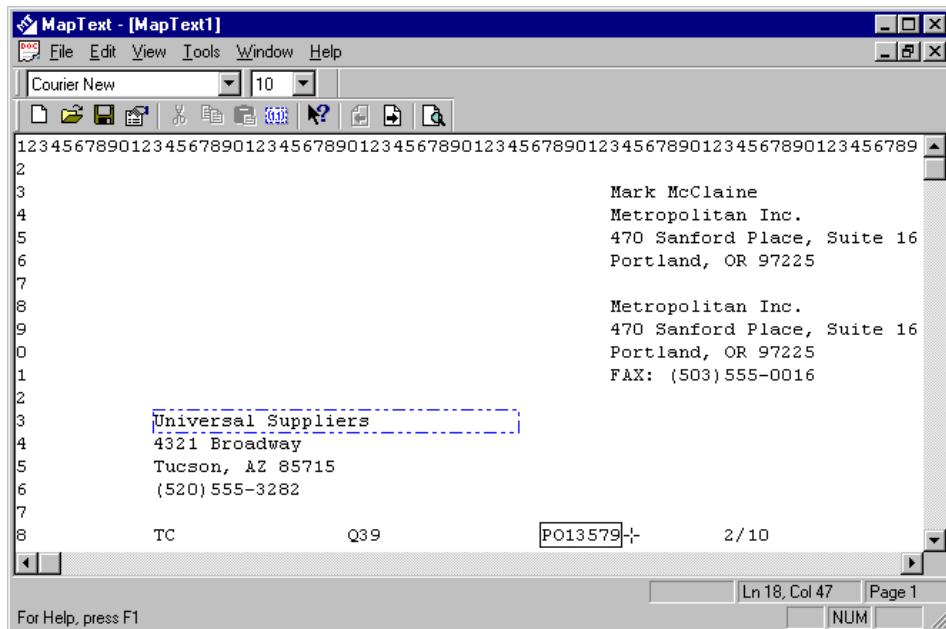
- Define the field in the document template that will contain the name of the recipient company. Drag the mouse to select the company name, for example **Universal Suppliers** of SampleData.txt. The **New Field** dialog box appears.

In the following example the field is longer than the company name it contains, allowing for longer company names in future documents.



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

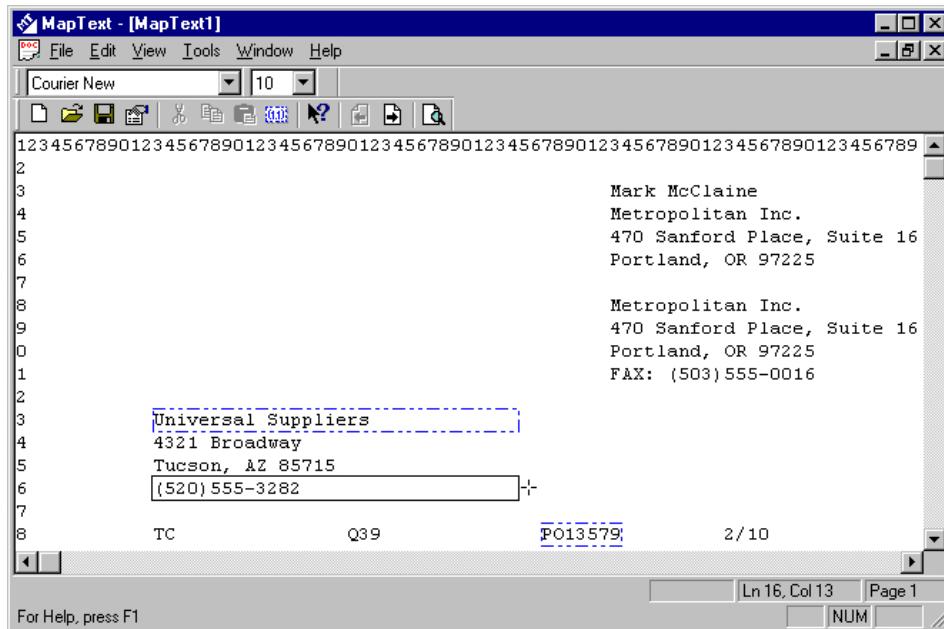
4. Note that the location and length of the field appear under **Cell location**.
 5. Click **OK**.
 6. Define the field in the document template that will contain the purchase order number. Drag the mouse to select the purchase order number, for example **PO 13579** of **SampleData.txt**. The **New Field** dialog box appears.



7. In the **Description** box, enter a description for the field, such as PO Number.
 8. 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 “[Field types](#)” on page [49](#).

9. Click OK.
 10. 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, for example **(520-555-3282)** of SampleData.txt. The **New Field** dialog box appears.



11. In the **New Field** dialog box, in the **Description** box, enter a description for the field, such as Fax Number.
12. In the **Field Type** list, click **Fax**.
13. Click **OK**.
14. On the **File** menu, click **Save**.
15. Give the filter template a descriptive name. It is saved as a MapText document (.mtd) in the OpenText Fax \Production\Include folder.

In this example, you have:

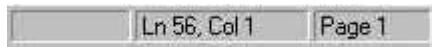
- Mapped the company name in purchase order documents to the “Company Name” field of the template. This created the FCL command {{company Universal Suppliers}} that OpenText Fax will use to address the fax.
- Mapped the purchase order number in purchase order documents to the “Purchase Order” field of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

When OpenText Fax 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.

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

To establish properties

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 created.
3. In the **Page Length (lines)** box, enter **56**.
4. `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).
5. 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.



6. 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 48.
7. Select **Portrait**.
8. 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 .

4.3.4 Step 4: Insert FCL

You can extend the filter template by adding FCL commands to set the proper syntax and the placement of the commands within documents. See “[FCL Commands](#)” on page 203.

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.
3. Enter the FCL using the proper syntax and delimiters.

4. To insert the commands 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** to close the Custom FCL dialog box. Click **OK** again to display the MapText window.
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.



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.

4.3.5 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 OpenText Fax Server can receive data from the host application and process the data correctly with the filter.

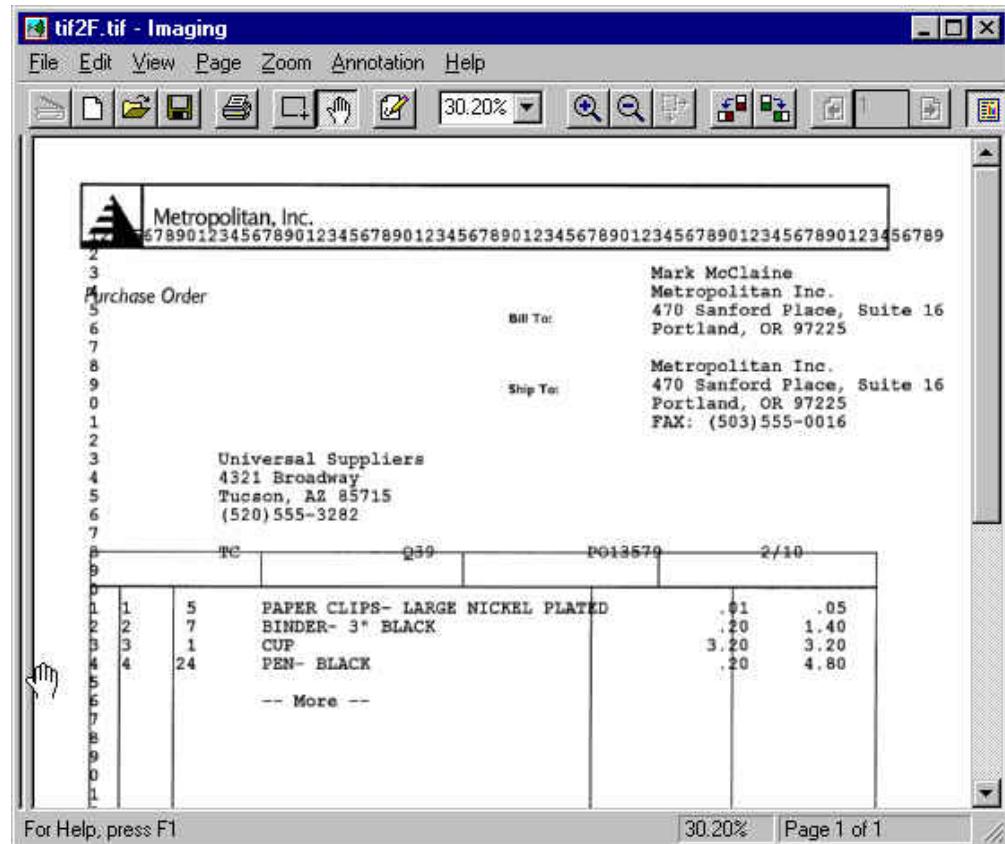
To preview the filter

1. 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.

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

In the following example, the form is not aligned with the document data. For help troubleshooting filter templates, see “[Troubleshooting](#)” on page 53.



4.4 Field types

The following table lists the field types and their uses.

Field type	Description	Map the field type to this data element
Abort	<p>Cancels the creation of a document when the specified text appears in the field.</p> <p>Corresponds to the {{abort}} FCL code.</p>	<p>This field type is commonly used for testing purposes.</p> <p>Example For testing, you might create documents in a large batch that contain the word "test" in the Abort field. The documents would not be sent.</p>
Comment	<p>Starts a new document when text in this field changes.</p>	<p>Map this field to data that is unique in each document.</p> <p>Example The purchase order number is unique in each document. When the purchase order number changes, a new document is started.</p> <p>See also the Page field type. The Page field type overrides the Comment field type.</p>
Company	<p>Specifies the recipient company name.</p> <p>Corresponds to the {{company}} FCL code.</p>	<p>Map this field to the name of the company that should receive the document.</p> <p>This information may be placed on the cover sheet or in a notification message.</p>
Contact	<p>Specifies the recipient name.</p> <p>Corresponds to the {{contact}} FCL code.</p>	<p>Map this field to the name of the person who should receive the document.</p> <p>This information may be placed on the cover sheet or in a notification message.</p>
Cover	<p>Specifies the cover sheet.</p> <p>Corresponds to the {{cover}} FCL code.</p>	<p>Map this field to the cover sheet file name.</p> <p>The cover sheet file must be stored in the directory OpenText Fax\Production\Covers. If the file is not found in the directory when the document is created, the default cover sheet is used.</p>

Field type	Description	Map the field type to this data element
Email	<p>Specifies the email address to send.</p> <p>Notifications about the sent document to the host application.</p> <p>The document via email using the InternetLink Module.</p> <p>Corresponds to the {{email}} FCL code.</p>	<p>The email address of the person or company who is sending this document (used to send a notification to the email address).</p> <p>The email address of the person or company to which you are sending this document (requires InternetLink).</p>
Fax	<p>Specifies the fax number.</p> <p>Corresponds to the {{fax}} FCL code.</p>	<p>Map this field to the fax number where the document should be sent.</p>
Graphic	<p>File name of a graphic.</p>	<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 <code>MarkJones.tif</code>.</p> <p>The graphic file format must be <code>.tif</code>. The file must be stored in the directory <code>OpenText Fax\Production\Forms</code>. If the file is not found in the directory when the document is created, then the image does 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 <code>OpenText Fax\Production\Include</code>. If the file is not found in the directory when the document is created, then an error occurs and the document is not processed.</p>

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 is used to send the document, included on the cover sheet, or included in a notification message.</p> <p>When you create a Lookup field, the Edit lookup file button appears. Click this button to create or edit a lookup file.</p> <p>The lookup table must be stored in the OpenText Fax\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 is printed on the cover sheet, and the fax number used to send the document.</p>
Owner	<p>Specifies the sender's name.</p> <p>Corresponds to the {{owner}} FCL code.</p>	<p>Map this field to the sender's name.</p> <p>The name can appear on the cover sheet or in notification messages.</p>
Page	<p>Starts a new document when the number 1 appears in this field</p>	<p>Map this field to the page number.</p> <p>See also the Comment field type. The Page field type overrides the Comment field type.</p>

Field type	Description	Map the field type to this data element
SendFax	<p>Use this field to print a document rather than fax it.</p> <p>The document is printed if N, a space, or null characters are found in this field.</p>	<p>Map this field to the fax number.</p> <p>This field type is used when not all recipients have a fax number or other address for transmission. Such documents are printed so that they could be mailed.</p>
TemplateID	Specifies the filter template to use to format the document.	<p>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.</p> <p>When text in the mapped field matches the text specified, the specified template is used to format the document.</p>
User1 User 2 User3	<p>Specifies user-defined information.</p> <p>Corresponds to the user-defined {{user1}}, {{user2}}, and {{user3}} FCL commands.</p>	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.	<p>Map this field to the phone number of the person who is receiving the document.</p> <p>The number can appear on the cover sheet or in notification messages.</p>

4.5 Troubleshooting

This section provides troubleshooting tips for building filter templates.

4.5.1 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.

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 47 .
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 47 .
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 47 .
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 39 .

4.5.2 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](#).

4.5.3 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](#).

4.5.4 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.

4.6 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 ([“Setting up a named pipe capture” on page 26](#))
- Directory scanning ([“Setting up directory scanning” on page 27](#))
- Serial capture ([“Setting up a serial capture” on page 29](#))
- TCP/IP port capture ([“Setting up a TCP/IP port capture” on page 32](#))

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

Command line option	Description
-c "filter makedoc"	<p>The option -c specifies one or more commands to execute on files received via the channel.</p> <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. <p>The pipe symbol (!) separates the two commands in the command line. The commands must be enclosed in quotation marks.</p>
-i <filename>.mtd	The option -i specifies that the specified MapText document (<filename>.mtd) should be included with the incoming data.
-f	Identifies <filename>.mtd 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 pipes the result to “makedoc” for processing.

Chapter 5

Reusing FCL commands across many documents

With the OpenText Fax Server, you can create a list of commands and/or text strings that can be used in many documents. This list is called an **include file**.

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. It can also contain a set of FCL commands that set the text size and font of the documents in which it is included.

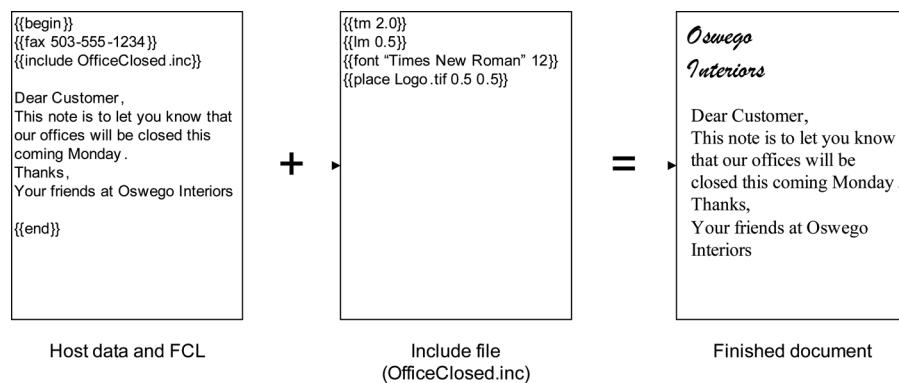
5.1 Include files

You have two options for creating and using include files:

- Global include files are provided in the `OpenText Fax\Production\Include` folder when you install the OpenText Fax 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.

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.



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.

5.1.1 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.

5.1.2 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 5-1: 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 5-2: Example: FCL Document with Global Include Files

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

Global.beg and global.end are used for general FCL commands that are used only one time 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 OpenText Fax 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.

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.

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.

5.2 Creating include files

You can use any word processor that produces 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 OpenText Fax\Production\Include folder.

5.2.1 Inserting content into global include files

To open any of the global include files in the OpenText Fax\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.

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, the file will attempt to insert itself in the FCL document. The resulting infinite loop will cause OpenText Fax to stop responding.

Every document sent through the OpenText Fax 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 determine into which global include file you should insert certain commands.

5.2.2 Storing include files

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

5.3 Linking include files

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

Use the {{include}} command to link an include file (one that you created and saved with an .inc extension in the OpenText Fax\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 OpenText Fax\Production\Include folder.



Note: Though valid include files must be stored with the extension .inc, you do not have to write the extension in the command itself. In the above example, either {{include OfficeClosed.inc}} or {{include OfficeClosed}} calls in the appropriate file.

For detailed information on the {{include}} command, including syntax and examples, see “[Creating FCL documents](#)” on page 161.

5.3.1 Linking global include files

The global include files are linked by default to every document that the OpenText Fax 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 6

Creating and attaching cover sheets and other files

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.

You can also attach other files at any point within a faxed document using FCL commands.

6.1 Cover sheets

OpenText Fax 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.

The OpenText Fax system supports two kinds of cover sheets:

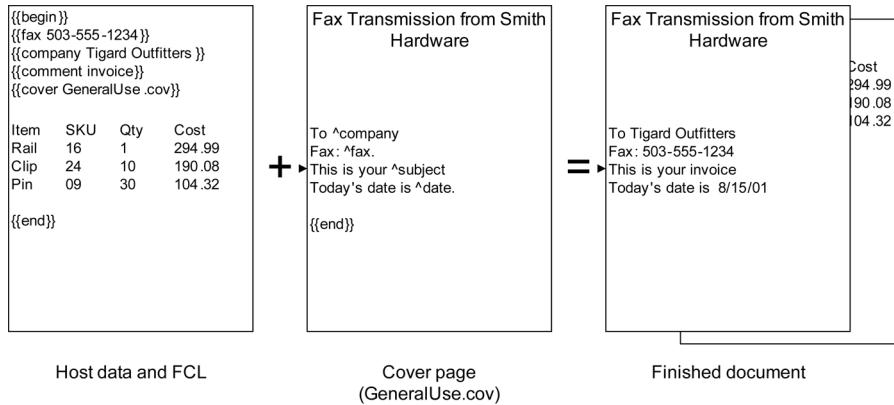
- **Production** cover sheets can contain FCL codes, text, and keywords. These cover sheets are attached to documents by inserting the {{cover}} command in the documents. The files are named with the extension .cov, and they are stored in the folder OpenText Fax\Production\Covers. The OpenText Fax Integration Module generates these cover sheets.



Note: Production cover sheets cannot be used when broadcasting. See “Using cover sheets in a broadcast” on page 167.

- **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 OpenText Fax\FCS. The OpenText Fax 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 *OpenText Fax - Administrator Guide (FXNET240400-AGD)*.

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



In this example, the command {{cover GeneralUse.cov}} retrieves the template file GeneralUse.cov from the OpenText Fax\Production\Covers folder. GeneralUse.cov contains text and keywords, which the OpenText Fax 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.

6.2 Creating cover sheet templates

Using default cover sheet templates

The OpenText Fax Integration Module includes a cover sheet called Auto.cov. This file is the default cover sheet, unless you specify a different one with the OpenText Fax Integration Module Configuration program. For information on establishing a default cover sheet, See ["Setting defaults for FCL documents" on page 162](#).

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 OpenText Fax\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.

6.2.1 Creating a basic cover sheet template

To create a basic cover sheet template

1. Open Microsoft Notepad or another word processing application that produces plain text.
2. Type text, FCL, or keywords in the template.
 - “Adding blocks of text to a cover sheet from the document data” on page 63
 - “FCL for cover sheets” on page 67
 - “Keywords for cover sheets” on page 64
3. Type {{end}} in the last line of the cover sheet.
4. Save the file with the extension .cov in the folder OpenText Fax\Production\Covers.

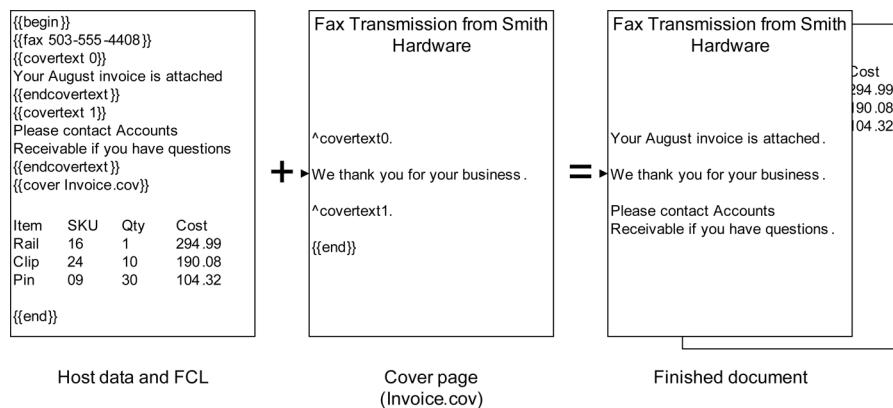
6.2.2 Adding blocks of text to a cover sheet from the document data

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.



6.2.3 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 (^).

See “[FCL Commands](#)” on page 203.

Keyword	FCL equivalent	Notes
<code>^altfax</code>	<code>{{altfax}}</code>	Includes whatever data is in the <code>{{altfax}}</code> code.
<code>^billing</code>	<code>{{billing}}</code>	Includes whatever data is in the <code>{{billing}}</code> or <code>{!billing}</code> code.
<code>^billing2</code>	<code>{{billing2}}</code>	
<code>^comment</code>	<code>{{comment}}</code>	Includes whatever data is in the <code>{{comment}}</code> code.
<code>^company</code>	<code>{{company}}</code>	Includes whatever data is in the <code>{{company}}</code> code.
<code>^contact</code>	<code>{{contact}}</code>	Includes whatever data is in the <code>{{contact}}</code> code.
<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 OpenText Fax 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.

Keyword	FCL equivalent	Notes
^dept	{{dept}}	Includes whatever data is in the {{dept}} code.
^docnum	N/A	The document number assigned by the OpenText Fax Integration Module.
^email	{{email}}	The recipient's email address.
^emailcc	{{email}}	The email address for an additional recipient.
^empid	{{empid}}	Includes whatever data is in the {{empid}} code.
^fax	{{fax}}	Includes whatever data is in the {{fax}} field. This uses the same information as the ^phone keyword.
^file	{{type file}}	The file name for {{type file}} documents.
^from	{{email}} {{from}}	
^localfax	N/A	Replaced with the local fax number set in the OpenText Fax Integration Module Configuration program.
^oecopies	{{onerror print}}	When {{onerror}} action is set to print, ^oecopies is replaced with the number of copies to be printed.
^onerror	{{onerror}}	Replaced with the {{onerror}} action specified for this document, or if none is specified, the default set in the OpenText Fax Integration Module Configuration program.
^onsuccess	{{onsuccess}}	The {{onsuccess}} action specified for this document. If none is specified, the default set in the OpenText Fax Integration Module Configuration program will be used.
^owner	{{owner}}	Includes whatever data is in the {{owner}} code.

Keyword	FCL equivalent	Notes
<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>	Includes whatever data is in the <code>{{fax}}</code> code. This uses the same information as the <code>^fax</code> keyword.
<code>^privatefax</code>	<code>{{privatefax}}</code>	
<code>^rti</code>	<code>{{rti}}</code>	If no <code>{{rti}}</code> string is specified, the <code>^rti</code> variable will use the default set in the OpenText Fax Integration Module Configuration program.
<code>^subject</code>	<code>{{comment}}</code> <code>{{subject}}</code>	
<code>^termid</code>	<code>{{termid}}</code>	
<code>^time</code>	<code>{{time}}</code>	If no <code>{{time}}</code> is specified, the current time will be used.
<code>^tranid</code>	<code>{{tranid}}</code>	
<code>^type</code>	<code>{{type}}</code>	
<code>^unique_id</code>	<code>{{unique_id}}</code>	Maps to the <code>unique_id</code> field in the OpenText Fax database.
<code>^user1</code>	<code>{{user1}}</code>	
<code>^user2</code>	<code>{{user2}}</code>	
<code>^user3</code>	<code>{{user3}}</code>	
<code>^voice</code>	<code>{{voice}}</code>	
<code>^winsecid</code>	<code>{{winsecid}}</code>	The OpenText Fax user ID of the originator of the fax.

6.2.4 Word-wrapping cover sheet note text

By default, text entered in the notes field of a cover sheet wraps at 70 characters.

To customize this setting

1. On the OpenText Fax 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.

6.2.5 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 “[Keywords for cover sheets](#)” on page 64.

See “[FCL Commands](#)” on page 203.

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 OpenText Fax Integration Module Configuration program. See “ Setting defaults for FCL documents ” on page 162.

Command	Description
{{email}}	Stores the return sender's return email address of the document. This command is sometimes also used to populate variables in notifications.
{{empid}}	Specifies the employee ID of the sender. This command is most often used to populated variables in notifications, but is sometimes also used with cover sheets.
{{owner}}	Specifies the name of the sender. This command is sometimes also used to populate variables in notifications.
{{privatefax}}	Specifies the private fax number of the sender.
{{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 OpenText Fax 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 fax originated. This is typically the name of the sending company. You can set a default name for the sending company in the OpenText Fax Integration Module Configuration program. See " Setting defaults for FCL documents " on page 162.
{{termID}}	Specifies the terminal identification from which the fax 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.

Command	Description
{{uniqueID}}	<p>Sets an identification number for each destination (fax number) within the document.</p> <p>This command is used most often for tracking. The OpenText Fax 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.</p> <p>Secondarily, this command is sometimes used in cover sheets and with notifications.</p>
{{user1}}, {{user2}}, {{user3}}	<p>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.</p>
{{userID}}	<p>Identifies the creator of this document.</p> <p>This command is most often used to populated variables in notifications, but is sometimes also used with cover sheets.</p>
{{voice}}	<p>Sets the voice number to be associated with the document.</p> <p>This command is most often used to populated variables in notifications, but is sometimes also used with cover sheets.</p>

6.3 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 OpenText Fax Integration Module would retrieve the file called FaxCoverPage.cov from the OpenText Fax\Production\Covers folder and attach it as the first page of the document.



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. If you create and attempt to attach a production cover sheet that is not saved with a .cov extension, errors will occur.

See “[FCL Commands](#)” on page 203.

6.4 Cover sheet logic

Because you can set a default cover sheet in the OpenText Fax Server Configuration program, set a default cover sheet for a OpenText Fax user and specify a cover sheet with the {{cover}} FCL command. The following table shows which cover sheet is used in different scenarios.

With these configuration settings			This cover sheet is used			
OpenText Fax user default cover sheet	OpenText Fax default cover sheet	Template in {{cover}} FCL command	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 FCL command in sent document				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 FCL command in sent document			X	
Cover.pcl	“none”	“ cover ” (.pcl, .doc, or .cov)	X			
Cover.pcl	“none”	“none”				X
Cover.pcl	“none”	“rfdefault”		X		
Cover.pcl	“none”	No FCL command in sent document				X

With these configuration settings			This cover sheet is used			
OpenText Fax user default cover sheet	OpenText Fax default cover sheet	Template in {{cover}} FCL command	The specified file name	The user default	The Integ. Module default	No cover sheet
Cover.pcl	Cover.cov	"cover" (.pcl, .doc, or .cov)	X			
Cover.pcl	Cover.cov	"none"				X
Cover.pcl	Cover.cov	"rfdefault"		X		
Cover.pcl	Cover.cov	No FCL command in sent document			X	
Cover.pcl	"rfdefault"	"cover" (.pcl, .doc, or .cov)	X			
Cover.pcl	"rfdefault"	"none"				X
Cover.pcl	"rfdefault"	"rfdefault"		X		
Cover.pcl	"rfdefault"	No FCL command in sent document		X		

6.5 Attaching other types of files

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, use the OpenText Fax InternetLink Module, described in ["Creating FCL documents with InternetLink commands" on page 187](#).

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 *OpenText Fax - Administrator Guide (FXNET240400-AGD)*.

6.5.1 Converting attached files

Attached files are converted to fax format (.tif) with the OpenText Fax conversion engine. If the conversion quality of faxed file attachments is low, the OpenText Fax server can be configured so that the applications associated with certain file formats are launched, and the native applications convert those documents to .tif. This is known as server-side application conversion, or "SSA."

To convert a file to fax format with SSA, the source application must be installed on the OpenText Fax server.

OpenText Fax attempts SSA conversions for all attachments. If the file type is not supported by SSA, or if the source application is not installed on the OpenText Fax server, then the conversion engine is used.

To set up the server for converting attachments to .tif, see *OpenText Fax - Administrator Guide (FXNET240400-AGD)*.

6.5.2 Storing attachments

For every attached file, you must either provide a path to the file in the {{attach}} command, or save the file in the OpenText Fax\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 OpenText Fax\Production\Forms.

The following example shows an FCL document that would attach a file called AugustInvoice.doc. The attachment would be converted to fax format, attached to the body of the main document at the start of a new page, and sent to (503) 555-0016.

For this example document to function properly, Microsoft Word must be installed on the OpenText Fax server computer (to use SSA to convert AugustInvoice.doc) and AugustInvoice.doc must already be saved in the OpenText Fax\Production\Forms folder.

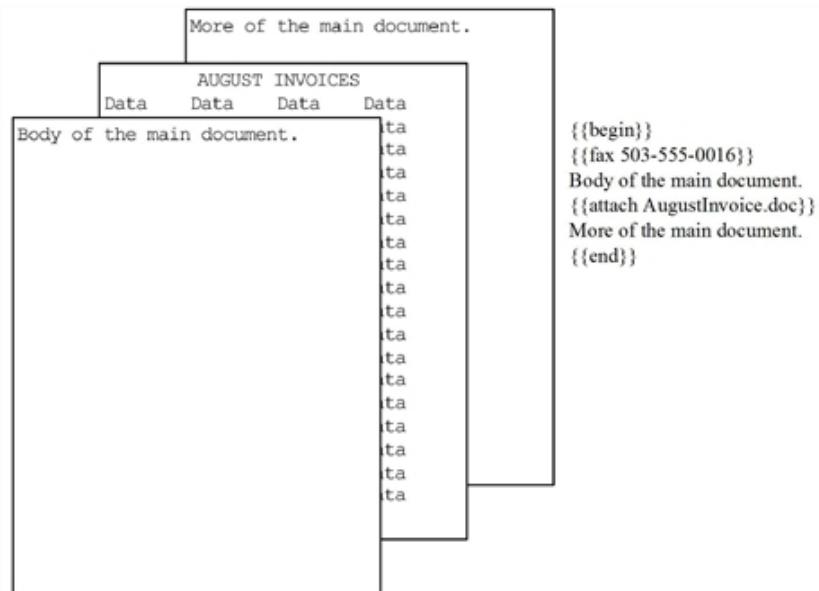


Figure 6-1: Attaching the file named AugustInvoice.doc

See “FCL Commands” on page 203.

Chapter 7

Defining fax page appearance

A series of FCL commands can be used to change the way a document looks. You can draw shapes, embed existing graphics, embed signatures, change margins and tabs, select specific fonts, or define page orientation.

Creating and configuring these shapes can supplement or replace background forms or graphics.

You can also change the appearance of a document with background forms. See “[Creating and linking background forms](#)” on page 170.

7.1 Drawing shapes

The following table lists the FCL commands for drawing shapes and provides a brief description. See “[FCL Commands](#)” on page 203.

Command	Description
<code>{{Box}}</code>	Draws a box whose size and location at specified coordinates. You can also place text inside the box.
<code>{{EndPoly}}</code>	Ends a polygon that is started with the <code>{{startpoly}}</code> command.
<code>{{FillBox}}</code>	Same as the <code>{{box}}</code> 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.
<code>{{Line}}</code>	Draws a line whose starting and ending points are specified by coordinates that you supply.
<code>{{LineTo}}</code>	Draws a line whose starting point is the current cursor location and whose ending point is specified by coordinates that you supply.
<code>{{LineWidth}}</code>	Lets you specify the width, in points, of the lines that you draw with other commands.

Command	Description
{{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.
{{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.

7.1.1 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.

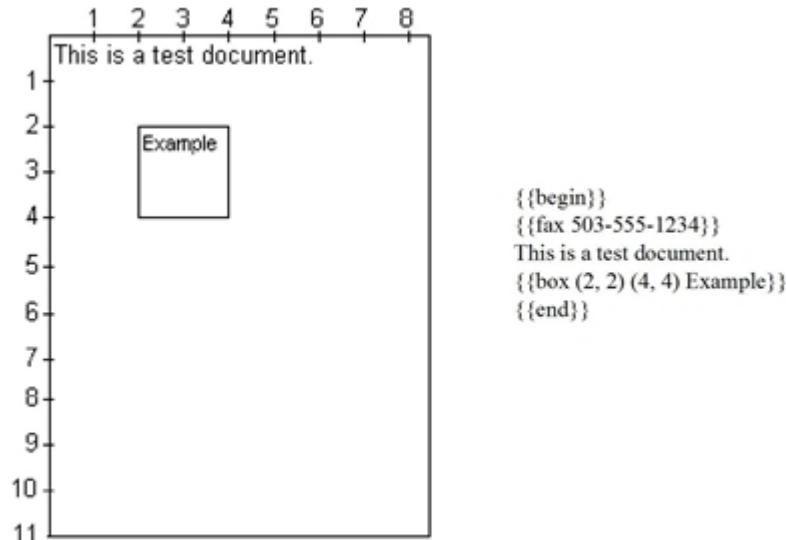
Example FCL code and finished documents are provided.

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 do not appear in the finished document.

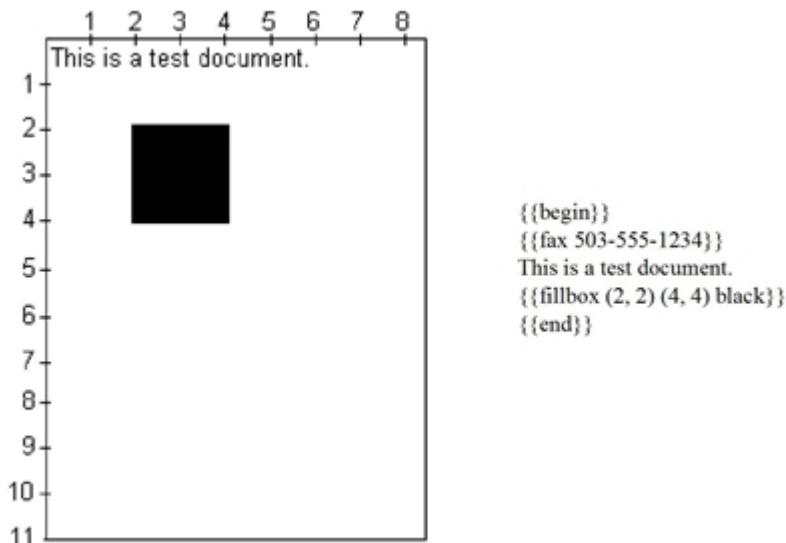
Box

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.



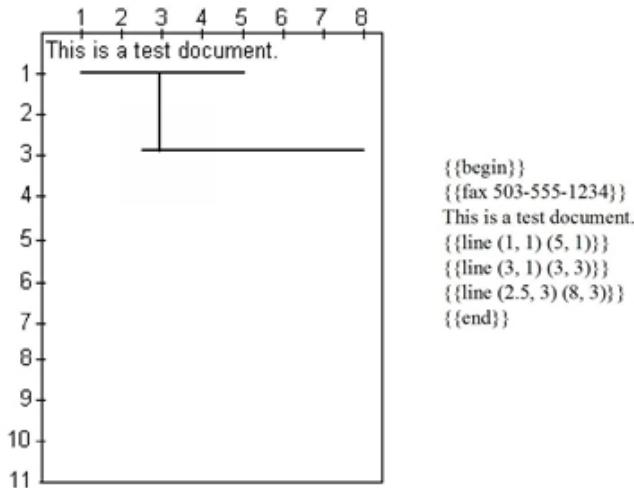
Fillbox

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.



Line

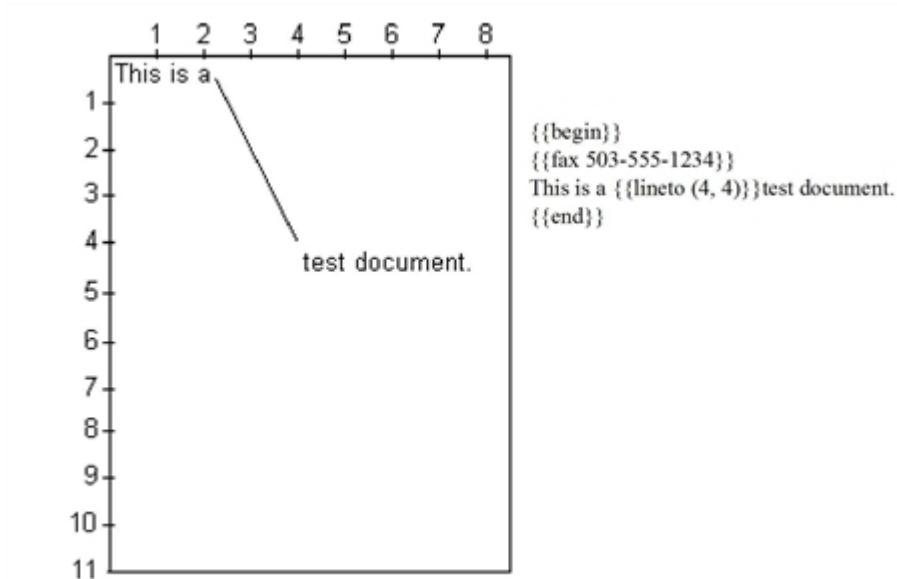
The line command draws a line between specified coordinates. The following example shows three {{line}} commands. In the first, (1, 1) are the x- and y-coordinates for the beginning of the line, and (5, 1) are the coordinates for the end of the line.



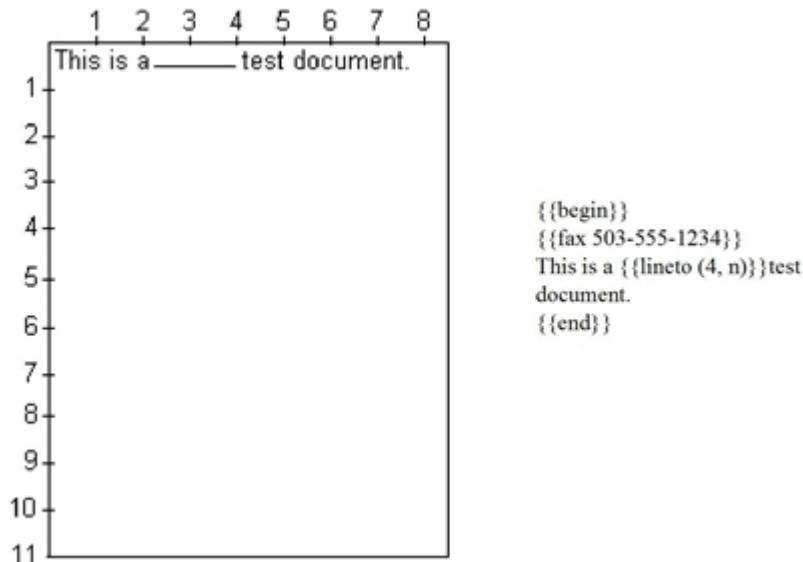
Lineto

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 OpenText Fax Integration Module finds the {{lineto}} command. In this case, the location is immediately after the phrase *This is a*. The line ends at the coordinates (4, 4). Then the remainder of the text appears.



The second example starts the line at the current location when the OpenText Fax Integration Module finds the {{lineto}} command. In this case, the location is immediately after the phrase *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; the position does not move down). Then the remainder of the text appears.



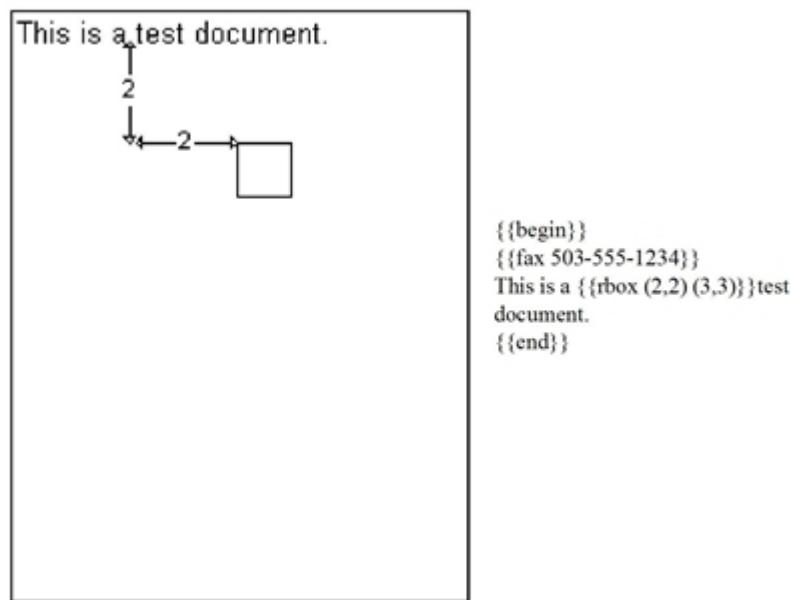
Rbox

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 `{fbox}` 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.

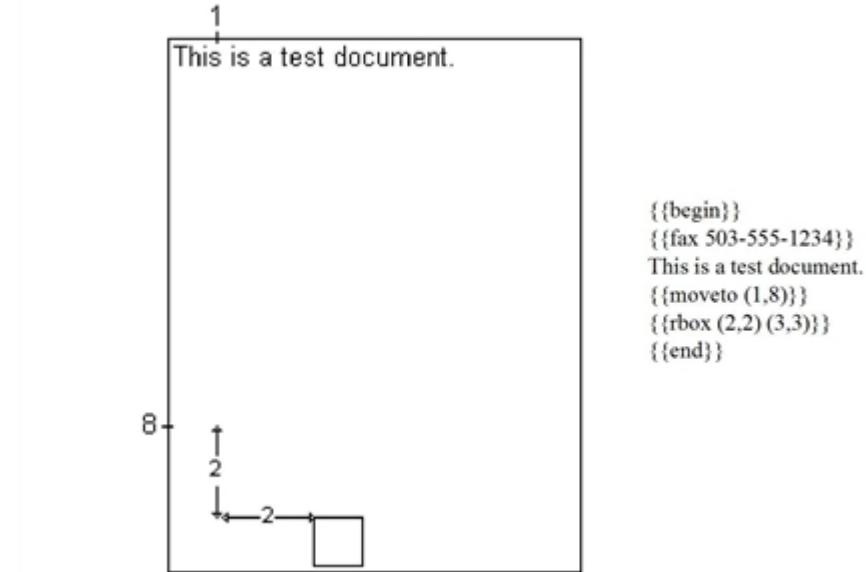
The first example starts the box at (2, 2) from the current location when the OpenText Fax Integration Module finds the `{rbox}` command. In this case, the location is immediately after the phrase *This is a*. The box ends at the coordinates (3, 3) from the current location. Then the remainder of the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



The next example starts the box at (2, 2) from the current location when the OpenText Fax Integration Module finds the `{rbox}` command. In this case, the location has been changed with the `{moveto}` command to (1, 8). 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.



7.2 Embedding signatures with FCL

You must create any signature files and save them as Class F TIF images. The Integration Module does not provide a method for creating these files. The most common method is to scan a signature. Regardless of method, the signature must be saved as a Class F TIF file.

7.2.1 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 OpenText Fax\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 OpenText Fax\Production\Forms.

7.2.2 Embedding a signature

To embed 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.

In the following example, the {{signature}} command establishes that DoctorRobertCribbs.tif is 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 OpenText Fax\Production\Forms folder.

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

Because it identifies a signature file, rather than placing it, the {{signature}} command can be used to manipulate the header of an FCL document, but not the body.

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 appears at the bottom of his or her letters.

See “[FCL Commands](#)” on page 203.

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 82.

7.3 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 OpenText Fax\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 (see “[Embedding signatures with FCL](#)” on page 81) 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.

7.4 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. See [Reusing FCL commands across many documents](#). Documents that contain a margin command do not use the defaults established in the global include file.

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.

7.4.1 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. See [“Setting defaults for FCL documents” on page 162](#) and [“FCL Commands” on page 203](#).

7.4.2 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. See [Reusing FCL commands across many documents](#) and “[FCL Commands](#)” on page 203.

7.4.3 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.

In the following example 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}}
```

7.5 Setting tabs with FCL

You can set tabs for individual documents or defaults to 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. See [Reusing FCL commands across many](#)

documents. Document that contains a tab command do not use the defaults established in the global include file.

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.

7.5.1 Setting tabs for individual documents

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

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.

Paragraph 1 of the document body.	<pre> {{begin}} {{fax 503-555-4489}} {{settab 0 1.5 L}} {{settab 1 2.5 L}} Paragraph 1 of the document body. {{tab 0 Paragraph 2 of the document body.}} {{tab 1 Paragraph 3 of the document body.}} {{end}}</pre>
1.5 tab → Paragraph 2 of the document body.	
2.5 tab → Paragraph 3 of the document body.	

7.5.2 Setting default tabs for all documents

To set 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.

See [Reusing FCL commands across many documents](#) and [“FCL Commands” on page 203](#).

7.6 Selecting and configuring fonts

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

Fonts can be either 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).

7.6.1 TrueType fonts

The OpenText Fax 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

See “[FCL Commands](#)” on page 203.

7.6.2 Support for other fonts

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

The OpenText Fax 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. Use the installed fonts or TrueType fonts for better font legibility.

7.6.3 Selecting fonts

You can establish a font for an individual document or a default font for all documents. See [Reusing FCL commands across many 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.

To change the default font for all documents, use the {{font}} command in a global include file.

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

7.6.3.1 Selecting fonts for individual documents

To set the font for a document, use the {{font}} FCL command in the FCL document. See “[FCL Commands](#)” on page 203.

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}}
```

7.6.3.2 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.

See [Reusing FCL commands across many documents](#) and [“FCL Commands” on page 203](#).

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

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

7.6.4 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 using the strikeout option within the {{font}} command.

See [“FCL Commands” on page 203](#).

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

<p>Table 1a</p> <p><u>Table 1b Body text, paragraph 1.</u></p> <p>Table 1c Body text, paragraph 2.</p> <p>Table 1d Body text, paragraph 3.</p> <p>Table 1e Body text in Lucida Sans, 12-point, italic. This is an installed font.</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>
--	---

Figure 7-1: To change the look of fonts with FCL

7.7 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 [Reusing FCL commands across many documents](#).

Documents that contain the {{orient}} command do not use the default in the global include file.

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.) See “[FCL Commands](#)” on page 203.

7.7.1 Setting page orientation for individual documents

To set orientation for a document, insert either {{orient portrait}} or {{orient landscape}} into the FCL document. See “[FCL Commands](#)” on page 203.

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}}
```

7.7.2 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.

See [Reusing FCL commands across many documents](#) and “[FCL Commands](#)” on page 203.

Chapter 8

Document transmission and notifications

8.1 Scheduling document transmission

You can use FCL commands to schedule when the OpenText Fax Integration Module sends documents. Unless you specify otherwise with one of these commands, the OpenText Fax Integration Module sends documents when it receives them from the host application.

The following table lists the scheduling commands and gives a brief description. See “[FCL Commands](#)” on page 203.

Table 8-1: The scheduling commands

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

Command	Description
{{priority}}	<p>Specifies a sending status of high, medium, 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 OpenText Fax 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 may be sent before the high priority batch documents are sent.</p>
{{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. 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 OpenText Fax Integration Module at 21:00 (9:00 p.m.) and you insert the command {{time 20:45}}, the document is sent immediately because 20:45 (8:45 p.m.) of the current day is in the past.</p>
{{UTC}}	Specifies a future date and time to send a document in universal coordinated time.

Chapter 9

Creating notification messages with FCL

You can configure the content of notification messages, how the Integration Module sends them, and how they are received.

They contain information such as whether a document was transmitted, any transmission errors, the date and time of transmission and the owner of the document.

Another form of notification is to print or fax a copy of the document to another recipient, such as a system administrator. See “[Setting up actions on document transmission](#)” on page 135.

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

When a document is received, generated, and transmitted from the OpenText Fax 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 .

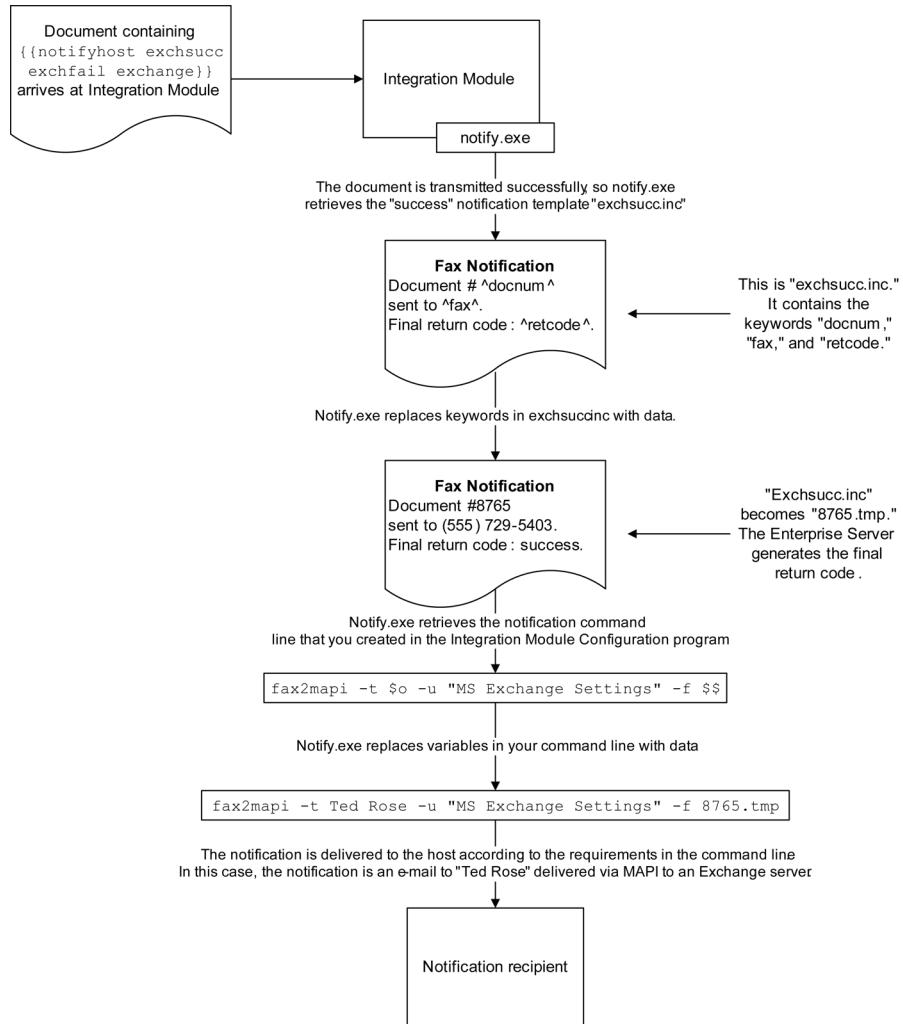
Notification messages are not generated by the Integration Module for incomplete faxes. Usually, these are faxes that are missing information that is required for sending (such as the fax number).

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 *OpenText Fax - Administrator Guide (FXNET240400-AGD)*).
- Use the OpenText Fax API to create custom notifications. See *OpenText Fax API Guide* for more information.

Example: The following illustration shows the FCL notification process for the following notification channel command line:

```
FAX2MAPI -t $o -u "MS Exchange Settings" -f $$
```



To create notifications for FCL documents

1. Create a template for the notification message (described in “[Creating notification templates](#)” on page 95). Notification templates determine the format and content of the notification message.
2. Create a notification channel (described in “[Creating notification channels](#)” on page 107). 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 in “[Including the {{Notifyhost}} command in documents](#)” on page 123). The {{notifyhost}} command specifies the notification template for the message and the notification channel to use.

Chapter 10

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, see [“Creating notification messages with FCL” on page 93](#). 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 is either the word “success” or a specific error code.

To create and save a notification template

1. Open Notepad or another text editing application that produces 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 96](#).
3. Save the file with the extension .inc in the OpenText Fax\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.

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

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

Keyword	FCL equivalent	Description
^altnfax^	{{altnfax}}	The alternate fax number.
^billing^	{{billing}}	Billing code 1.
^billinfo2^	{{billing2}}	Billing code 2.
^comment^	{{comment}} {{subject}}	The subject appears in the subject field of the email message. If no subject appears in the document, the default subject is "Success Notification" or "Error Notification."
^company^	{{company}}	The destination company name.
^completiontime^	N/A	The date and time when the fax transmission was completed.
^contact^	{{contact}}	The recipient name.
^csi^	{{csi}}	If no CSID (call subscriber ID) is specified, the default set in the OpenText Fax Integration Module Configuration program will be used.
^date^	{{date}}	If no date is specified, the default is the date the fax was submitted.
^dept^	{{dept}}	The string defined as department.
^docnum^	N/A	The document number assigned by the Integration Module.
^duration^	N/A	The duration of the fax call in seconds.

Keyword	FCL equivalent	Description
<code>^email^ or ^mailto^</code>	<code>{{email}}</code> <code>{{to}}</code>	The keyword <code>^email^</code> is the recipient address. With other notifications, <code>^email^</code> is typically the email address of the originator of the document.
<code>^emailfrom^</code>	<code>{{from}}</code>	The email address of the originator of the document. If no email address appears in the document, the default is postmaster@domain .
<code>^emailcc^</code>	<code>{{cc}}</code>	The email address where a copy of the message should be sent.
<code>^emailsubject^</code>	<code>{{subject}}</code>	The email subject.
<code>^empid^</code>	<code>{{empid}}</code>	The recipient employee ID.
<code>^fax^</code>	<code>{{fax}}</code>	The recipient fax number. Same as <code>^phone^</code> .
<code>^phone^</code>	<code>{{fax}}</code>	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 <code>^faxstatus^</code> and <code>^statustype^</code> keywords” on page 101 .
<code>^humantranstime^</code>	N/A	The date and time the fax was transmitted in human-readable format.
<code>^numretry^</code>	N/A	The total number of fax attempts.
<code>^owner^</code>	<code>{{owner}}</code>	If no <code>{{owner}}</code> command appears in the sent document, the default is the OpenText Fax user name of the owner of the fax.
<code>^ownerid^</code>	<code>{{owner}}</code>	The OpenText Fax user 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.

Keyword	FCL equivalent	Description
<code>^replyto^</code> or <code>^reply_to^</code>	<code>{{replyto}}</code>	The reply-to name.
<code>^retcode^</code>	N/A	The text of the code returned from the fax board after transmission. This is either "success" or a specific error message. See "The <code>^retcode^</code> and <code>^statusstring^</code> keywords" on page 102 .
<code>^rti^</code>	<code>{{rti}}</code>	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 <code>^retcode^</code> and <code>^statusstring^</code> keywords" on page 102 . For a detailed description of the status, see error and status messages in the <i>OpenText Fax - Administrator Guide</i> (FXNET240400-AGD)).
<code>^statustype^</code>	N/A	Numeric return code; see "The <code>^faxstatus^</code> and <code>^statustype^</code> keywords" on page 101 .
<code>^termid^</code>	<code>{{termid}}</code>	The termination code.
<code>^time^</code>	<code>{{time}}</code>	If no time is specified, the default is the time the fax was submitted.
<code>^tranid^</code>	<code>{{tranid}}</code>	The transmission ID generated by the OpenText Fax server.
<code>^transtime^</code>	N/A	The date and time that the fax was transmitted. To specify the format for the date and time, see "The <code>^completiontime^</code> and <code>^transtime^</code> keywords" on page 99 .

Keyword	FCL equivalent	Description
^type^	{{type}}	The document transmission type (fax, email, 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 email or certified email instead. These options require the InternetLink Module (for email or mime documents) or the SecureDocs module (for certified email documents).
^unique_id^	{{unique_id}}	The unique ID for the fax assigned by the OpenText Fax server.
^user1^	{{user1}}	User-defined data code 1.
^user2^	{{user2}}	User-defined data code 2.
^user3^	{{user3}}	User-defined data code 3.
^userid^	{{userid}}	OpenText Fax user ID.
^voice^	{{voice}}	The recipient's phone number.
^winsecid^	{{winsecid}}	The OpenText Fax user ID of the originator of the fax.

10.2 The ^completiontime^ and ^transtime^ keywords

The ^completiontime^ keyword and variables provide the date and time that the fax transmission was completed. The ^transtime^ keyword and variables provide the date and time that the document was transmitted.

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

Table 10-1: Example: ^Transtime^ Variables with Results

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

Command and variable	Result
<code>^transtime %Y-%m-%dTime%H:%M:%S^</code>	2013-08-09 Time 12:40:31

The following table lists the variables that can be used with the `^completiontime^` and `^transtime^` keywords.

Variable	Description
<code>%a</code>	Abbreviated weekday name.
<code>%A</code>	Full weekday name.
<code>%b</code>	Abbreviated month name.
<code>%B</code>	Full month name.
<code>%c</code>	Date and time representation appropriate for locale.
<code>%d</code>	Day of month as a digit (01-31).
<code>%H</code>	Hour in 24-hour format (00-23).
<code>%I</code>	Hour in 12-hour format (01-12).
<code>%j</code>	Day of year as a digit (001-366).
<code>%m</code>	Month as a digit (01-12).
<code>%M</code>	Minute as a digit (00-59)
<code>%p</code>	Current locale's A.M./P.M. indicator for 12-hour clock.
<code>%S</code>	Second as a digit (00-59).
<code>%T</code>	Local time zone designation (TZD).
<code>%U</code>	Week of year as a digit, with Sunday as first day of week (00-51). If <code>^completiontime^</code> or <code>^transtime^</code> begins with <code>%U</code> , the time displayed is UTC rather than local.
<code>%w</code>	Weekday as a digit (0-6; Sunday is 0).
<code>%W</code>	Week of year as decimal number, with Monday as first day of week (00-51)
<code>%X</code>	Date representation for current locale
<code>%x</code>	Time representation for current locale.
<code>%y</code>	Year without century, as a digit (00-99).
<code>%Y</code>	Year with century, as a digit.
<code>%z</code>	Time zone name or abbreviation; no characters if time zone is unknown.
<code>%Z</code>	Time zone name or abbreviation; no characters if time zone is unknown.

Variable	Description
%%	Percent.
#	Prefixes formatting code to modify as follows: %#c – long date and time %#x – long date %#d, %#H, %#I, %#j, %#m, %#M, %#S, %#U, %#w, %#W, %#y, %#Y – remove any leading zeros.

10.3 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 appear in the notification message.

^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.

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

Table 10-2: ^Faxstatus^ codes

Code	Status
0	Fail
2	OK

10.4 The ^retcode^ and ^statusstring^ keywords

The following table shows the values for the ^retcode^ and the ^statusstring^ keywords.

^retcode^	^statusstring^	Description
Success	OK	OpenText Fax successfully sent the fax over the phone lines. This does not guarantee successful receipt of the fax at its destination.
Busy	Fax number busy	OpenText Fax tried to send the fax the required number of times (default=5 times) and each time the phone number was busy.
Transmission error	Transmission error (Tx/Rx error)	Text for Tx/RX error: The receiving fax machine was experiencing difficulties or excessive line noise.
Poor quality	Poor quality transmission	There was too much line noise for the fax to be transmitted correctly. The fax may have been sent, but it may be difficult to read.
No answer	No answer at fax number	The fax machine at the receiving end did not answer or a wrong number was dialed.
Bad FCS information	Bad cover sheet information	Incorrect information was entered on the cover sheet using the Fax Information screen or embedded codes.

^retcode^	^statusstring^	Description
Conversion error	Problem converting fax body or cover sheet	Error occurred during conversion of body or FCS.
FCS text creation error	Problem creating cover sheet	Error occurred during creation of FCS text.
Unable to schedule	Problem scheduling fax	Scheduling failed for some reason.
Human answered	Human answered the phone	OpenText Fax detected an unknown sound after it dialed the fax number. The unknown sound could be a human voice, recording, or line noise.
G2 fax machine	A non-group 3 fax device answered	You have attempted to send a fax to a Group II machine, rather than a Group III machine.
Local in use	Line in use	OpenText Fax was not able to send the fax because there were no outgoing phone lines available.
Phone line problems	Phone line problem	There is a problem with the phone lines at the fax server.
Invalid formtype	Invalid form type specified	The specified form does not exist or is corrupt in the New Fax dialog box or through embedded codes.
Invalid signature	Invalid signature specified	Invalid signature used.
No authorization	No authorization for attempted signature usage.	No authorization for attempted signature usage
Discarded	Fax blocked by board.	The fax was discarded
Invalid phone number	Invalid characters in phone number	The phone number contained invalid characters.
Invalid billing code	Invalid billing code	
Invalid embedded code	Invalid embedded code	
OCR operation failed	OCR processing failed	
Print operation failed	Print processing failed	
Unauthorized user	No authorization to create new library document	
Fax number blocked by dialing rule	Fax blocked from dialing phone number	One or more dialing rules disallow sending faxes to the specified phone number.

^retcode^	^statusstring^	Description
Unknown	Unknown error reported by fax board	
Unknown	Email send error	

10.5 Sample notification templates

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 sent
 {{end}}
```

The following table shows typical notification template files and the resulting notification messages.

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
Date ^transtime %B %d, %Y^	Date May 22, 2001
Sent to this number: ^fax 40^ at ^transtime %H:%M^	Sent to this number: 503-555-1234 at 8:37
Regarding ^comment 40^	Regarding Inv. # 12345
Sent to ^contact 40^	Sent to John Smith
Sending confirmation: ^retcode 20^	Sending confirmation: success

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

Chapter 11

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.

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.

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

This notification 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 email 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 recognizes.
- Variables that supply information about the sent document for the notification message.

To create a new notification channel

1. On the OpenText Fax server on the **Start** menu, click **OpenText Fax**, and then click **OpenText Fax Enterprise Fax Manager**.
2. In the left pane, expand the server on which the Integration Module is running, and then click **Services**.
3. In the list, right-click **OpenText Fax Integration Module**, and on the shortcut menu, click **Configure Service**. The **Integration Module Configuration** window opens.
4. Right-click **Notification Channels**, and on the shortcut menu, click **Add Output Device**. The available settings for the notification channel appear in the right pane.
5. In the **Name** box, enter a descriptive name for the channel. The channel number is assigned as you add channels.
6. In the **Command line** box, enter the case-sensitive command line for the channel.



Note: You can use the **Command line** arrow to add the commands.

For information on the command line to use for each type of notification, see:

- “[Sending notifications via SMTP](#)” on page 111
 - “[Sending notifications to a database](#)” on page 113
 - “[Sending notifications to an SMS-enabled device](#)” on page 114
 - “[Sending notifications via 3270 emulation](#)” on page 116
 - “[Sending notifications via FTP](#)” on page 117
 - “[Sending notifications via IBM WebSphere MQ](#)” on page 118
 - “[Sending notifications to Notes](#)” on page 120
 - “[Sending notifications to Microsoft Exchange](#)” on page 121
7. 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.
 8. Click **OK** to save the new notification channel.

11.1 Setting the number of retries for notifications

For each channel that exists for the program `Notify.exe`, you can configure the number of notification attempts. If a notification fails after the set number of retries, an error is logged and the failed notification file is saved in the `~\OpenText Fax\Production\Error` directory so that it can be reconciled.

To set the number of notification retries for a channel

1. Use a tool such as `regedit.exe` to edit the Windows registry. For each channel for which to set the number of retries, create the following registry key: `HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\RightFax\Production\Notify\ n`. Replace `n` with the number of the notification channel.
2. Create the DWORD key name **AttemptsToTryNotification**, and enter the number of retries in the **Datafield**. The default is 10 retries, and retries occur at 45 second intervals.

11.2 Creating notification channels in a Shared Services environment

Using the same notification channels on all servers in a Shared Services environment ensures that notifications are sent through the correct channels.

To create and distribute notification channel settings

1. Configure the integration modules on one server, including the notification channel settings.
2. Open the **Registry Editor**, and export and save the following registry key: HKLM\SOFTWARE\Wow6432Node\RightFax\Production\Notify
3. Import the registry to the other servers.

11.3 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 OpenText for information on other available notification methods.

Notification methods and executable program files

`Notify.exe` on the OpenText Fax 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.

Connection method	Program that sends the notification	Description
SMTP	<code>Mailsend.exe</code>	Sends messages via SMTP to an SMTP-compliant mail server.
ODBC database	<code>Dbnotify.exe</code>	Sends notifications to an ODBC data source.
SMS	<code>Rfsmss.exe</code>	Sends notifications via SMS.
3270 emulation	<code>Hlpisend.exe</code>	Sends notifications back through 3270; uses high-level-language application-programming interface (HLLAPI). This requires third-party 3270 emulation software.
FTP	<code>Ftpit.bat</code>	Is a Perl script used to log on to an FTP server and transfer files.

Connection method	Program that sends the notification	Description
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.

Notification channel variables

For each of the notification methods, one or more command line parameters support variables that pull information from the FCL in the document that is sent. The following table lists the command line variables that can be used with all notification methods.

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 is used in the command when it executes.

Variable	FCL equivalent	Notes
\$\$	N/A	File name of the notification that is created by Notify.exe.
\$^	{{termid}}	The termination code of the sent fax.
\$B	{{billing}} {{billing2}}	Billing codes 1 and 2.
\$C	{{comment}}	Comment text.
\$c	{{emailcc}}	The email address where a copy of the message should be sent.
\$d	N/A	The document number assigned by the Integration Module.
\$D	{{dept}}	The string defined as department.
\$E	{{empid}}	The employee ID.

Variable	FCL equivalent	Notes
\$f	{{from}}	The email address of the originator of the document.
\$o	{{owner}}	The OpenText Fax 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 email subject.
\$t	{{to}}	The email address of the recipient of the message.
\$T	N/A	The TIF file name of the document.
\$U	N/A	The unique ID of the fax. The unique ID cannot contain a colon. This variable can be used to name the files that are created by <code>Notify.exe</code> .
\$w	{{winsecid}}	The OpenText Fax 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.

11.4 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 11-1: Mailsend.exe command line options

Option	Description
-a	Abort if connect fails.

Option	Description
<code>-c address</code>	<p>Email address where a copy of the message will be sent. The email address can appear in the following formats:</p> <p>Ashutosh Apte <AshutoshApte@opentext.com></p> <p><AshutoshApte@opentext.com></p> <p>AshutoshApte@opentext.com</p> <p><code>Mailsend.exe</code> will convert these formats to <AshutoshApte@opentext.com>.</p>
<code>-f address</code>	Email address of the sender of the document.
<code>-h</code>	Displays online help for <code>Mailsend.exe</code> .
<code>-H</code>	Input file contains mail headers.
<code>-m "mailhost"</code>	The name of the mail host where messages will be sent. The name must be surrounded by quotation marks.
<code>-o</code>	Obtain the recipient email address from the first line of the file, and obtain the subject of the email message from the second line.
<code>-s subject</code>	Subject of the email message.
<code>-taddress</code>	<p>The email address of the recipient of the notification message. The email address can appear in the following formats:</p> <p>Ashutosh Apte <AshutoshApte@opentext.com></p> <p><AshutoshApte@opentext.com></p> <p>AshutoshApte@opentext.com</p> <p><code>Mailsend.exe</code> converts these formats to <AshutoshApte@opentext.com>.</p>
<code>-v</code>	Display verbose messages.

```
mailsend -m "smtpserver.yourhost.com" -t "$o" -f
OpenTextFaxAdmin@company.com -s "Notification of Fax" $$
```

Element3	Description
<code>mailsend</code>	The name of the executable file that will process the notification.
<code>-m "smtpserver.yourhost.com"</code>	The name of the mail host.

Element3	Description
-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 OpenTextFaxAdmin@company.com	The email system's address and the sender's SMTP address, which is required to receive the notification.
-s "Notification of Fax"	The text that will appear in the subject line of the email message.
\$\$	The file name of the document.

11.5 Sending notifications to a database

To create a notification channel that sends SQL messages to a database

1. Set up a new Data Source Name (DSN): In **Control Panel**, under **System and Security**, click **Administrative Tools**, and then **Data Sources (ODBC)**. The **ODBC Data Source Administrator** dialog box opens. Click **Add** and complete the options to create the new DSN. Make a note of the DSN name as you 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^')
```

3. Configure a new notification channel (described in [“Creating notification channels” on page 107](#)) using the **Dbnotify.exe** program in the **Command Line** box.

Syntax

```
dbnotify { -f <fileName> | -s SQL } [ -u UserID ] [ -p Password ] DSN
```

Dbnotify.exe command line options

-f<fileName>

Fully qualified file name of a file containing the SQL script for the notification. You must use -f or -s, but not both.

-s<SQL>

SQL script to execute containing the notification script. You must use -f or -s, but not both.

-u<UserID>

User ID. This is only required if your ODBC data source was not configured with a user name.

-p<Password>

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
```

Element	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.

11.5.1 Verifying that the database notification was sent

By default, DBNotify returns a value that reflects how many rows of the table were updated. Zero indicates that the database notification failed.

To customize the return value, use `regedit.exe` to edit the Windows registry. For each notification channel, create the following registry key: `HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\RightFax\Production\Notify\n`. Replace n with the number of the notification channel.

Create the REG_SZ key name `SuccessfulReturnValue` and enter the desired success value in the **Data** field.

11.6 Sending notifications to an SMS-enabled device

To send notifications via SMS

1. Subscribe to and configure the OpenText Fax SMS Service. See *OpenText Fax - Administrator Guide (FXNET240400-AGD)*.
2. Create notification templates for both success and failure notifications (described in “[Creating notification templates](#)” on page 95). 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).
3. Configure a new notification channel (described in “[Creating notification channels](#)” on page 107) using the `Rfsms.exe` program in the **Command Line** box.

Syntax

```
rfsms -s<SMSService> -d<SMSNumber>
[ {-m<File>| "<MessageText>" } ] [-v] [-f<RFServer>] [-u<UserID>] [-p<Password>]
```

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 11-2: Rfsms.exe command line options

Option	Description
- s <ServiceID>	The service ID of the SMS service you configured in OpenText Fax.
- d <SMSNumber>	The phone number of the SMS device where the message will be sent.
{- m<File> "<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.
[- f<RFServer>]	The name of the OpenText Fax 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.
[- u<UserID>]	The OpenText Fax user ID required to log on to the OpenText Fax server. This is only required if you are not using a trusted account.
[- p<Password>]	The password for the OpenText Fax user ID specified with the -u option.

Example:

```
rfsms -fRFServer -uAdministrator -pPassword -sSMS1 -d$1 -m$$
```

Element	Description
rfsms	The name of the executable file that will process the notification.
-fRFserver	The name of the OpenText Fax server on which the specified SMS service resides.
-uAdministrator	The OpenText Fax user ID used to log on to the server.
-pPassword	The password for the specified user ID.

Element	Description
-sSMS1	The service ID of the SMS service in OpenText Fax that sends SMS messages.
-d\$1	Maps to the {{SMS}} FCL code.
-m\$\$	The file containing the alert text to send.

11.7 Sending notifications via 3270 emulation

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

Syntax

```
hlpisend [ options ]filename
```

Table 11-3: Hlpisend.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 <code>Hlpisend.exe</code> .
-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.
-v	Display verbose messages.
-w seconds	Specify wait after input.

Example:

```
hlpisend -s A -H 2 -l login.inc $$
```

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.

11.8 Sending notifications via FTP

To create a notification channel that sends messages via FTP, you must first create the FTP command line by using the Integration Setup Wizard. Subsequent changes to the command line or new FTP notification channels can be made using the Integration Module Configuration program. To start the Integration Setup Wizard, see “[Using the Integration Setup Wizard](#)” on page 22.

The program `Ftpit.bat` uses a Perl script to log on to an FTP server and transfer files.



Note: To use `Ftpit.bat`, download and install a copy of `Perl.exe` from <http://www.perl.org> (<http://www.perl.org/>). There is no charge for Perl.

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 11-4: Ftpit.bat command line options

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

Example:

```
ftpit yourhost.com franklins qwerty $$
```

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.

11.9 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

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

Table 11-5: Mqput.exe command line options

Option	Description
<code>-C channel</code>	Name of the IBM WebSphere MQ channel to use for connection.
<code>-H hostname</code>	Fully qualified domain name of the IBM WebSphere MQ queue manager.
<code>-M queue manager</code>	
<code>-Q queue</code>	IBM WebSphere MQ queue to retrieve messages from.
<code>-T</code>	Enable TLS.  Note: You must set the Queue Channel SSL Authentication to Optional .
<code>-S " <SSL CipherSpecs>"</code>	Specifies the SSL CipherSpecs of the queue channel SSL settings. For example: ANY_TLS12_OR_HIGHER  Note: You must set the Queue Channel SSL Authentication to Optional .

Option	Description
<code>-R "<SSL key repository>"</code>	Specifies the SSL key repository of the queue manager SSL configuration settings. For example: C:\ProgramData\IBM\MQ\qmgrs\QM_APPLE\ssl\key\key  Note: You must set the Queue Channel SSL Authentication to Optional .
<code>-d</code>	Display debugging output. This is helpful if you experience difficulty connecting to the server.
<code>-e days</code>	Days until message expires. Default is no expiration.
<code>-f</code>	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.
<code>-i input</code>	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.
<code>-p port</code>	TCP/IP port number to use for remote connection. The default is 1414.
<code>-r priority</code>	Message priority. The default is 0.
<code>-s</code>	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.
<code>-v</code>	Display version information.
<code>-1</code>	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.
<code>-tCCSID</code>	Specifies the codeset name for a language. A list of the codeset IDs (CCSIDs) supported by WebSphere MQ is available from IBM.

`mqput -C RF_Chан -H qmmaster2 -M RightFax -Q RF_Notify -i $$ -p 1414 -1`
where the elements are:

Element	Description
<code>mqput</code>	The name of the executable file that will process the notification.
<code>-C RF_Chан</code>	The IBM WebSphere MQ channel name.

Element	Description
-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 \$\$	
-p 1414	The TCP/IP port number.

11.10 Sending notifications to Notes

To create a notification channel that sends messages to a Notes system using the Notes API

1. Configure the notification channel as described in this guide.
2. Configure the OpenText Fax Notes Email Gateway. Refer to *OpenText Fax - Notes Module Administrator Guide (FXNET240400-AIN)*.
3. In Windows Control Panel advanced system settings, set the path environment variable to include the path to the nNotes.dll.
4. Restart the OpenText Fax server.

Syntax

fax2note [options]recipient

Table 11-6: Fax2note.exe command line options

Option	Description
-t emailaddress	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 email message.
-a	Identifies a file to attach to the email message.
-f	Identifies an ASCII text to use as the body of the email message.
-o	Identifies the owner of the email message.

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

Element	Description
fax2note	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.

11.11 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 11-7: Fax2mapi.exe command line options

Option	Description
-a file name	Attach the specified file.
-f file name	Name of an ASCII text file to use as the message body.
-i file name	Name of an ASCII text file to use as the message body.
-k	Keep sent mail.
-p password	Profile password.
-q CSID	CSID associated with the FAX address type of recipient. This option overrides any address specified with -t.
-r DID	DID associated with the FAX address type of recipient. This option overrides any address specified with -t.
-s subject	Subject line of the email message.
-t email address	Email 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.

Option	Description
-u profile	Mail profile name.
-z	File (in RTF format) to be used as a message body.

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

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

Chapter 12

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 OpenText Fax 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 a channel number (from 1 to 128) or name. If no channel is specified, the default channel, number 16, will be used.

Syntax

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

```
{{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 will send 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. See [Reusing FCL commands across many documents](#).

12.1 Creating the {{Notifyhost}} command

The {{notifyhost}} command must contain:

- The name of the success template, as described in ["Creating notification templates" on page 95](#).
- The name of the failure template, as described in ["Creating notification templates" on page 95](#).
- The notification channel number or name, as described in ["Creating notification channels" on page 107](#).

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}}
```

12.2 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. More detailed information on each of the commands is also contained at the end of this document. See “[FCL Commands](#)” on page 203.

For a comparison of the commands listed here and the keywords that you can use in notification templates, see “[Template file keywords](#)” on page 96.

Command	Description
<code>{{Billing}}</code>	The billing code of the document owner. This command is sometimes used to populate variables in cover sheets.
<code>{{Billing2}}</code>	A secondary billing code of the document owner. This command is sometimes used to populate variables in cover sheets.
<code>{{Comment}}</code>	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.
<code>{{Company}}</code>	The company name for the document. This command is most often used to populate variables in cover sheets but is sometimes used with notifications.

Command	Description
{{Contact}}	<p>The contact name for the document.</p> <p>This command is most often used to populate variables in cover sheets but is sometimes used with notifications.</p>
{{CSI}}	<p>The CSID. This is usually the general fax number for the enterprise.</p> <p>You can set a default CSID in the Integration Module Configuration program. “Setting defaults for FCL documents” on page 162.</p>
{{Dept}}	The string defined as department.
{{Email}}	The email address for the recipient of the notification message.
{{EmpID}}	<p>The employee ID of the fax owner.</p> <p>This command is sometimes used to populate variables in cover sheets.</p>
{{Owner}}	<p>The document owner’s name.</p> <p>This command is most often used to populate variables in cover sheets but is sometimes used with notifications.</p>
{{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 OpenText Fax XML Interface. ReplyTo is the field in the submit post that the OpenText Fax XML Interface populates to determine where to send the notification.
{{TermID}}	<p>The ID of the terminal from which the document originated.</p> <p>This command is sometimes used to populate variables in cover sheets.</p>
{{TranID}}	<p>The ID of the transaction that produced the document.</p> <p>This command is sometimes used to populate variables in cover sheets.</p>

Command	Description
<code>{{UniqueID}}</code>	An identification number for each destination (fax number) within the document. This command is used most often for tracking. The Integration Module generates 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.
<code>{{User1}}</code> <code>{{User2}}</code> <code>{{User3}}</code>	User-defined data, such as the originator of the document (person, group, or other information).
<code>{{UserID}}</code>	The OpenText Fax user ID of the creator of this document. This command is sometimes used to populate variables in cover sheets.
<code>{{Voice}}</code>	A voice telephone number. This command is sometimes used to populate variables in cover sheets.

Chapter 13

Testing that the host application is correctly receiving notifications

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 section 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 93.

The following procedure requires that you have created a notification channel, a notification template, and the {{notifyhost}} command identifying the channel and template.

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 OpenText Customer Support.

To test the host

1. Send a test document from the host to the Integration Module .
2. 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.
3. Verify that you receive a notification from the Integration Module in the location that you specified when you created the notification channel.
4. 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 notification messages](#)” on page 131.

13.1 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).

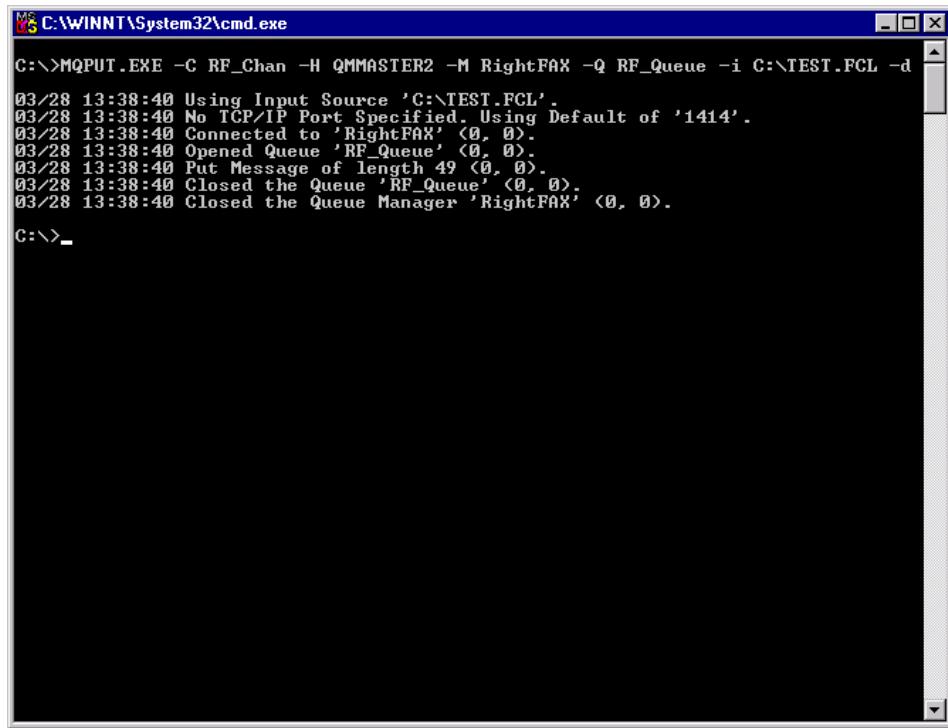
This procedure requires that you have correctly configured:

- 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 118).

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.
Command line options are case-sensitive. You must enter the command here exactly as you did in the **Add Output Device** dialog box. Otherwise errors will occur.
3. Press **Enter**.

If the notification channel connection is successful, you will see messages similar to that shown in the following example. If the output connection is not successful, you will see error messages.



The screenshot shows a Windows Command Prompt window titled "C:\WINNT\System32\cmd.exe". The command entered was "MQPUT.EXE -C RF_Chан -H QMMASTER2 -M RightFAX -Q RF_Queue -i C:\TEST.FCL -d". The output log shows the following sequence of events:

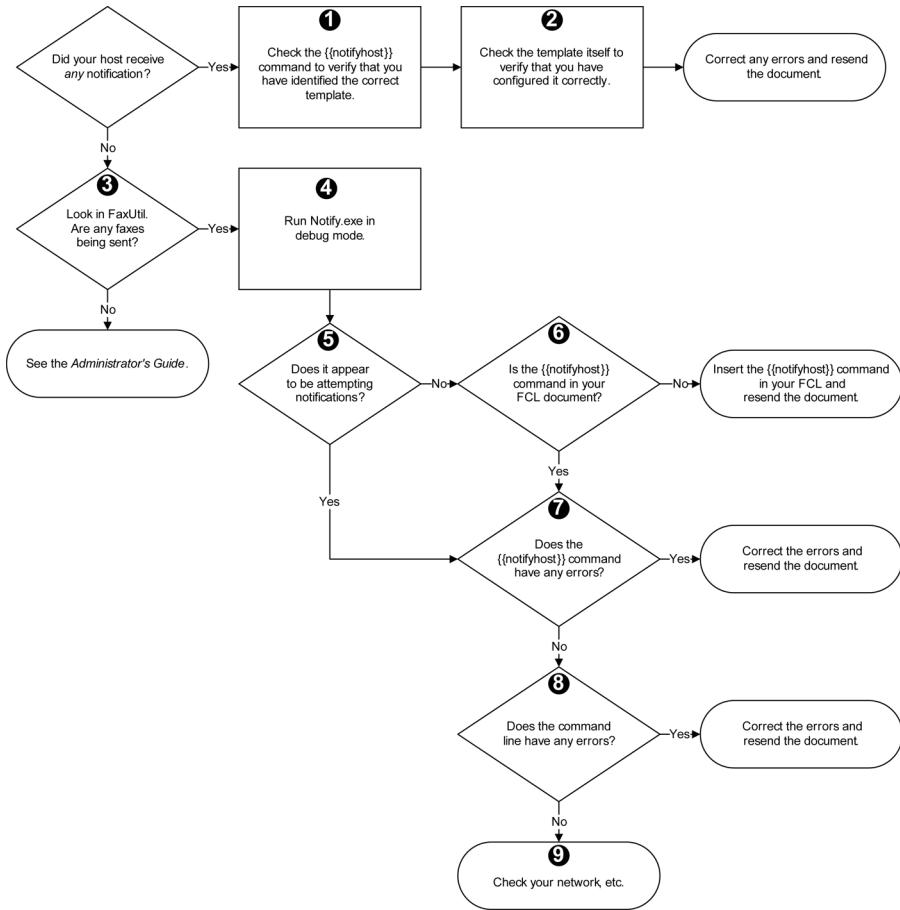
```
C:\>MQPUT.EXE -C RF_Chан -H QMMASTER2 -M RightFAX -Q RF_Queue -i C:\TEST.FCL -d
03/28 13:38:40 Using Input Source 'C:\TEST.FCL'.
03/28 13:38:40 No TCP/IP Port Specified. Using Default of '1414'.
03/28 13:38:40 Connected to 'RightFAX' <0, 0>.
03/28 13:38:40 Opened Queue 'RF_Queue' <0, 0>.
03/28 13:38:40 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:\>
```


Chapter 14

Troubleshooting notification messages

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



1. If you created several templates with similar names, a minor typographical error can identify the wrong template, resulting in a notification that contains errors and non-populated 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, see ["Template file keywords" on page 96](#).

3. When you do not receive a notification after sending a document, first verify that the OpenText Fax 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.
- Change directory to `C:/Program Files (x86)/OpenText Fax/Production/bin`.
- 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, information appears in the command prompt window describing 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. Check the connection (see "[Configuring the Integration Module to receive data](#)" on page 25).

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 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` returns 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` returns a message such as "Unable to include file: <path>.inc").
- A notification ID that does not exist (`Notify.exe` returns 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) because `Notify.exe` uses the default ID (16) when no ID exists. The default action for ID 16 is to copy to NUL.

If you leave ID 16 unchanged, then files that are being copied to NUL probably have no ID associated with them.

If you changed the action for ID 16, 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 (“[Creating notification channels](#)” on page 107). Common problems include:
 - A misspelled or invalid executable file, or (if you specified a path to the executable) the wrong path. (`Notify.exe` returns 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` returns 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` does 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.

Chapter 15

Setting up actions on document transmission

You can configure the OpenText Fax Integration Module to notify you when a document has been sent or has errors in transmission.

Notifications can be in several formats, such as sending the notification to a printer or sending a copy of the fax to a specified user. These actions are similar to notification messages. See “[Creating notification messages with FCL](#)” on page 93.

Notification messages contain information such as whether a document was transmitted, any transmission errors, the date and time of transmission and the owner of the document.

15.1 Sending notifications of successful and failed transmissions

You can set the following default actions to be performed when a document is transmitted successfully (success action) or cannot be sent (error action): For each action type you can enter a recipient to receive a copy of the document, either as a notification of successful transmission or a notification of a failure to transmit. These defaults are set in the Integration Module Configuration.

To override the default actions, use the following FCL commands:

- The {{onsuccess}} FCL command in a document will override the default for the success action.
- The {{onerror}} FCL command in a document will override the default for the error action.

15.1.1 Setting the default success and error actions

To set the default success and error actions

1. On the Start menu, click **OpenText Fax**, and then click **OpenText Fax Enterprise Fax Manager**.
2. In the **Fax Servers** list, click the name of the server on which the OpenText Fax Integration Module is running.
3. In the **Service Name** list, double-click **OpenText Fax Integration Module**.
4. In the left pane of the **Integration Module Configuration** window, under **General**, click **More Options**.
5. Under **Fax Options**, do any of the following:

- To notify a recipient of successful transmission by sending a copy of the document, select the **Fax on Success** check box. In the **To Number** box, enter the destination fax number.
 - To notify a recipient of failed transmission by sending a copy of the document, select the **Fax on Error** check box. In the **To Number** box, enter the destination fax number.
6. Under **Delete Options**, specify when you want the fax images deleted from the server:
- **Never** – Fax images are never automatically deleted.
 - **On Success** – Only fax images that have been sent successfully are automatically deleted.
 - **Always** – Fax images are automatically deleted both after successful and failed transmission attempts.
7. Click **OK**.

15.1.2 Overriding the default success or error action

To override the default for the success action

1. Add the {{onsuccess}} command to the FCL.
2. You can specify destinations of *delete*, *certified*, *fax*, *email*, *mime*, or *nothing* with the {{onsuccess}} FCL command. See “[FCL Commands](#)” on page 203.

In the following example, a copy of the document will be faxed to another recipient if it is successfully transmitted.

```
 {{begin}}
 {{onsuccess fax 503-555-1234}}
 {{fax 503-555-4489}}
 Body of the document to be sent.
 {{end}}
```

To override the default for the error action

1. Add the {{onerror}} command to the FCL.
2. You can specify destinations of *delete*, *certified*, *fax*, *email*, *mime*, or *nothing* with the {{onerror}} command. See “[FCL Commands](#)” on page 203.

In the following example, a copy of the document will be faxed to another recipient if it is not successfully transmitted.

```
 {{begin}}
 {{onerror fax 503-555-1234}}
 {{fax 503-555-4489}}
 Body of the document to be sent.
```

{{end}}}

15.2 Performing actions on documents with FCL errors

When the OpenText Fax Server processes a document, it extracts data from the FCL commands, then creates and sends the document. You can specify which actions the system should take if there is an error that prevents the document from transmitting.

To set default actions

1. On the **Start** menu, click **OpenText Fax**, and then click **OpenText Fax Enterprise Fax Manager**.
2. In the **Fax Servers** list, click the name of the server on which the OpenText Fax Integration Module is running.
3. In the **Service Name** list, double-click **OpenText Fax Integration Module**.
4. In the left pane of the **Integration Module Configuration** window, under **Processes**, click **FCL Processor**.
5. To print the document, select **Print FCL to printer**, and then enter the name of the printer as defined in the Enterprise Fax Manager.
6. To fax the document, select **Fax FCL to number**, and enter the fax number.
7. 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 “[Creating notification messages with FCL](#)” on page 93.

The document can be sent to Microsoft Exchange as an email message. Therefore, in the **Execute command line** box, enter:

```
fax2mapi -t john.doe@company.com -f $$ -u "MS Exchange Settings"
```

In this example, when a document fails because of any FCL error, the OpenText Fax Integration Module will send it to the email address john.doe@company.com.

Chapter 16

Programming for the OpenText Fax XML Interface

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



Note: For the submission to the XML interface to be successful, the XML must be properly formatted and all XML entities properly escaped.

16.1 Introduction to the OpenText Fax 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 OpenText Fax schemas ([“The schemas” on page 149](#)).

The OpenText Fax API for Java is for Java programmers. It provides an alternate method of creating and sending XML to the OpenText Fax server. The API for Java allows access to XML Interface functionality without requiring that a customer know XML or the OpenText Fax XML Interface schemas. See [“Programming for the OpenText Fax API for Java” on page 175](#).

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

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

16.2 Installing the XML Interface

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

16.2.1 Minimum system requirements

In addition to the minimum system requirements for the OpenText Fax server and OpenText Fax Integration Module, Microsoft Internet Information Server (IIS) version 7.0 or later must be installed on the OpenText Fax server.

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.

16.2.2 Installing the XML Interface

Follow the instructions for installing the OpenText Fax server in the *OpenText Fax - Installation guide (FXNET240400-IGD)* and use the following specific steps:

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

After the OpenText Fax 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 OpenText Fax 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 OpenText Fax server. You can leave this option blank if you will not be using SMTP to deliver OpenText Fax alerts and notifications.

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

By default, the CGI-exe and ISAPI-dll Handler Mappings are disabled.

Additional XML development tools are located in the \Program Files\OpenText Fax \Production\XML folder.

To enable the Module mapping for rfwebcon.dll

1. In IIS manager, select the **rFXML** virtual directory.
2. In the Features View under IIS, double-click **Handler Mappings**.
3. Right-Click under Enabled and select **Add Module Mapping**.
4. Enter the following information in the **Add Module Mapping** window:

- **Request path:** rfwebcon.dll
 - **Module:** Isapi
 - **Module Executable:** <path to rfwebcon.dll>
 - **Name:** rfwebcon
5. Click **OK**. A warning appears asking if you want to make this change. Click **Yes**. An IIS reset should not be required.

16.3 XML Interface functions

Note that not every function is available for each transport method. For more information, see “[Introduction to the OpenText Fax XML Interface](#)” on page 139.

16.3.1 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 OpenText Fax 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 document ID for each recipient’s document or letting the OpenText Fax 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 OpenText Fax 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 OpenText Fax XML schema (see “[The schemas](#)” on page 149). 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 OpenText Fax schema. Or, you can create an XSLT to convert the document to FCL.

The companion to **XML_FAX_SUBMIT** is **XML_FAX_SUBMIT_REPLY**, which informs you of each unique document ID (if you did not create your own). 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 OpenText Fax 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 which is more detailed than **XML_FAX_SUBMIT_REPLY**.

16.3.2 Query

To determine the status of any document in the OpenText Fax 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 OpenText Fax 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.

16.3.3 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 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.

16.3.4 Notification

To receive a notification that a document was or was not sent successfully from the OpenText Fax 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 [Document transmission and notifications](#).

16.4 Transports

Note that some transport methods do not support all of the OpenText Fax XML Interface functions. For more information, see “[Introduction to the OpenText Fax XML Interface](#)” on page 139.

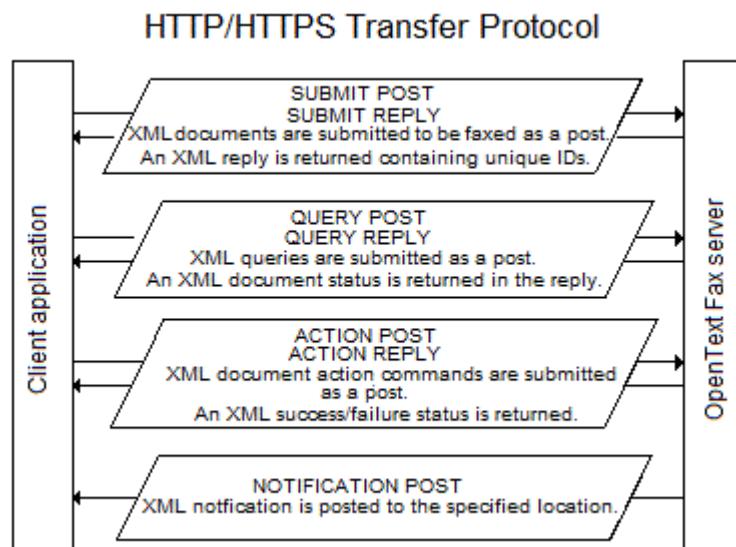
You can provide transport and access list security using standard mechanisms. Use HTTPS instead of HTTP to achieve transport encryption. The Web server providing the RFWebCon.dll resource can also generate user, IP address, and domain restrictions.

The XML Interface software supports three transport methods: HTTP or HTTPS, FTP, and IBM WebSphere MQ. You can also use the OpenText Fax API for Java to create and send XML via HTTP. See “[Programming for the OpenText Fax API for Java](#)” on page 175.

16.4.1 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 OpenText Fax server. All XML Interface functions are supported.

Each HTTP/S post is formatted with the library name of the 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.



16.4.2 Example HTTP or HTTPS post

Call	Response
POST /RFWebCon.DLL HTTP/1.1	HTTP/1.1 200 OK
Host: www.opentext.com	Content-Type: text/xml
Content-Type: text/xml	Content-Length: nnnn
Content-Length: nnnn	X-Captaris-Method: submit_reply
X-Captaris-Method: submit	
<XML_FAX_SUBMIT>...	</XML_FAX_SUBMIT>
</XML_FAX_SUBMIT>	

16.4.3 HTTP or HTTPS 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.

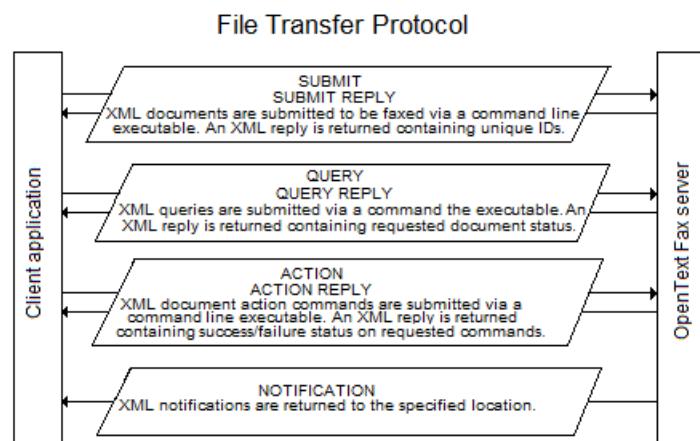
16.4.4 HTTP or HTTPS notifications

To receive notifications in this manner, you must provide an HTTP listener. The application `Postfile.exe` can be used to send messages from the OpenText Fax server, but it needs the path to the HTTP listener for you to actually receive the message.

16.4.5 File transport

File transport for XML allows an XML document to be processed using a command line executable (`Parsexml.exe`). When combined with existing OpenText Fax 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.



Command line syntax

```
parseXML [options] <xmlfile> [attachments]
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.

16.4.6 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

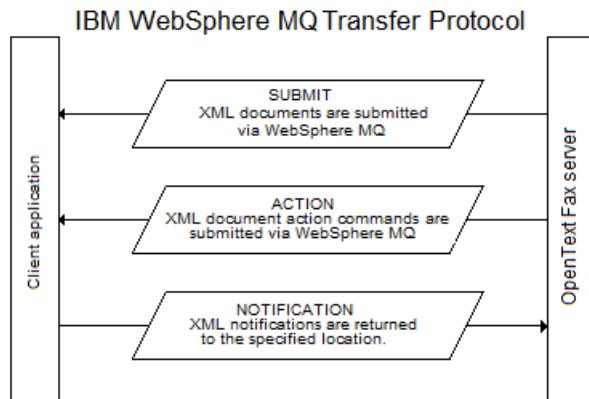
When attachments are included inline, they significantly increase the size of the XML message. Some XML parser implementations load the entire XML message into memory, which causes performance degradation when large messages are processed. An alternative to inline attachments is to include them as separate files.

In the following example command line, po.doc and list.pdf are attached files:

```
parseXML submit.xml po.doc list.pdf
```

16.4.7 IBM WebSphere MQ transport

IBM WebSphere MQ acts as an intermediary in connecting the host application to the OpenText Fax server software. It offers multiple connection options and requires some advanced configuration. For more information, see the *IBM's WebSphere MQ documentation*.



16.4.8 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].

16.5 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 you can also use formatted text such as Rich Text Format (.rtf) or binary files such as Microsoft Word (.doc) documents.

16.5.1 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.

If multibyte characters appear in text-only attachments, XML attachments may fail to convert correctly. If so, you can edit the Windows registry.

To support multibyte characters in text-only attachments

1. On the OpenText Fax server, open the Windows registry and browse to HKLM\Software\RightFax\Production.
2. Create the key Parse\Propagate and add the DWORD value BOMToBeginctv.
 - To insert any byte order mark (BOM) from the text file into attachments that are added with the BEGINCVT command, set the value to 1.
 - To not insert any byte order mark (BOM) from the text file into attachments that are added with the BEGINCVT command, set the value to 0.
3. Close the Windows registry.

16.5.2 Using formatted text data

To use formatted text formats such as Rich Text Format

1. Use the **type** attribute, setting it to the extension associated with that document type.
2. Copy the data into the XML node.

The following example is for a cover sheet in RTF:

```
<ATTACHMENT>
<COVER_TEXT type="RTF">
{\rtf1\ansi\ansicpg1252\deff0\deflang1033{\fonttbl{\f0\fswiss\fcharset0 Arial;}}
{*\generator Msftedit 5.41.21.2500;} \viewkind4\uc1\pard\f0\fs20 This is the text that
is printed on the cover. \par}
</COVER_TEXT>
</ATTACHMENT>
```

16.5.3 Using binary data

To use binary data, such as Word documents (.doc), Acrobat files (.pdf), or graphic images (.jpg, .bmp, etc.)

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 BASE64.
4. Copy the encoded data into the XML node.

In the following example, the body text is in Microsoft Word (.doc) format and the ellipses (...) represent the many pages that are contained in an actual Word file:

```
<ATTACHMENT>
<BODY type="DOC" encoding="BASE64">
OM8R4KGxGuEAAAAAAAAAAAAAAAPgADAP7/CQAGAAAAAAAAAAAAABAAAABAAA
EAAWAAAAAEAAAD+///AAAAAFUAAAD...
aW9uIG9mIGhvdyB0byBkbyB0aGlzLiB0b3RlIHRoYXQgdGhpcyBpcyBqdXN0IHRoZSBiYXNpY3Mu
DUNyZWFOZSBhbIBSRmF4U3VibW10IG9iamVjdC4NU2VOIHRo
...
...
...
UmlnaHRGQVggc2VydmVyLCB1c2luZyBzZXRUYXJnZXRVUkwgbWW0aG9kLiBUaG1zIHRha2VzIGEg
</DATA>
</ATTACHMENT>
```

16.6 The schemas

The schemas listed in this section define the structure and type of content that the OpenText Fax 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 OpenText Fax schema. Or, you can create an XSLT to convert the document to FCL. For sample documents, see “[Sample documents based on the schemas](#)” on page 156.

16.6.1 XML_FAX_SUBMIT_schema

For a sample document based on this schema, see “[XML_FAX_SUBMIT](#)” on page 156.

```
<?xml version="1.0" encoding="UTF-8" ?>
- <!--
rfxml version="2.0.1"
-->
- <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="PRIVATEFAX" 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="USER1" model="closed" content="textOnly" dt:type="string" />
<ElementType name="USER2" model="closed" content="textOnly" dt:type="string" />
<ElementType name="USER3" 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" />
<element type="USER1" minOccurs="1" maxOccurs="1" />
<element type="USER2" minOccurs="1" maxOccurs="1" />
<element type="USER3" minOccurs="1" maxOccurs="1" />
</ElementType>
- <!--
DESTINATION STUFF
-->
- <!--
Common Recipient elements and attributes
-->
```

```
-->
<AttributeType name="unique_id" dt:type="string" />
<AttributeType name="pdftype" 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">
<attribute type="unique_id" />
<attribute type="pdftype" />
<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">
<attribute type="unique_id" />
<element type="TO_EMAIL" minOccurs="1" maxOccurs="1" />
<element type="CC_EMAIL" minOccurs="0" maxOccurs="1" />
<element type="SUBJECT" 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="BCC_EMAIL" minOccurs="0" maxOccurs="1" />
<element type="INCLUDE_INC" 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">
<attribute type="unique_id" />
<element type="PRINTER_NAME" minOccurs="0" maxOccurs="1" />
<element type="COPIES" 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">
<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="r4" />
<AttributeType name="lm" dt:type="r4" />
<AttributeType name="bm" dt:type="r4" />
<AttributeType name="font_name" dt:type="string" />
<AttributeType name="font_size" dt:type="i1" />
<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_size" />
<attribute type="font_leading" />
<attribute type="font_pitch" />
</ElementType>
- <ElementType name="FONT" content="empty">
<AttributeType name="name" dt:type="string" required="yes" />
<AttributeType name="size" dt:type="i1" />
<AttributeType name="leading" dt:type="i1" />
<AttributeType name="pitch" dt:type="i1" />
<attribute type="name" />

```

```
<attribute type="size" />
<attribute type="leading" />
<attribute type="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>
```

16.6.2 XML_FAX_SUBMIT_REPLY.dtd

For an example document based on this dtd, see “[XML_FAX_SUBMIT_REPLY](#)” on page 158.

```
<!ELEMENT XML_FAX_SUBMIT_REPLY (FAX+)>
<!ATTLIST FAX unique_id CDATA #REQUIRED>
<!ELEMENT FAX (RETURN_CODE, MESSAGE)>
<!ELEMENT RETURN_CODE (#PCDATA)>
<!ELEMENT MESSAGE (#PCDATA)>
```

16.6.3 XML_FAX_QUERY_schema

For an example document based on this schema, see “[XML_FAX_QUERY](#)” on page 158.

```
<?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>
```

16.6.4 XML_FAX_QUERY_REPLY.dtd

For an example document based on this dtd, see “[XML_FAX_QUERY_REPLY](#)” on page 158.

```
<!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)>
```

16.6.5 XML_FAX_ACTION_schema

For an example document based on this schema, see “[XML_FAX_ACTION](#)” on page 159.

```
<?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>
                <AttributeType name = "docid" dt:type="string"/>
                <ElementType name="XML_FAX_ACTION" content="eltOnly">
                    <element type="FAX" minOccurs='1' maxOccurs='*' />
                    </ElementType>
                </ElementType>
            </ElementType>
        </Schema>
```

16.6.6 XML_FAX_ACTION_REPLY.dtd

For an example document based on this dtd, see “[XML_FAX_ACTION_REPLY](#)” on page 159.

```
<!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)>
```

16.6.7 XML_FAX_NOTIFICATION.dtd

For an example document based on this dtd, see “[XML_FAX_NOTIFICATION](#)” on page 159.

```
<!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)>
```

16.7 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 149.



Note: For example documents that require a name space (such as `<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.

16.7.1 XML_FAX_SUBMIT

```
<?xml version="1.0"?>
<!--rfxml version="2.0.1" -->
<XML_FAX_SUBMIT stylesheet="XML_FAX_SUBMIT.XSL" xmlns="x-schema://schemas/XML_FAX_SUBMIT_schema.xml">
  <SEND_DATE_TIME>2000-01-24T15:20:00-08:00</SEND_DATE_TIME>
  <INCLUDE_BEG>xm1.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>
    <USER1>This is user1 field</USER1>
    <USER2>This is user2 field</USER2>
    <USER3>This is user3 field</USER3>
  </SENDER>
  <DESTINATIONS>
    <FAX unique_id="PRODXML:0001">
      <TO_FAXNUM>555-1111</TO_FAXNUM>
      <INCLUDE_INC>xm1.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.csv</COVERSHEET>
      <INCLUDE_DEF>xm1.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@company.com</TO_EMAIL>
      <CC_EMAIL>fred@company.com</CC_EMAIL>
      <BCC_EMAIL>wilma@company.com</BCC_EMAIL>
    </EMAIL>
    <PRINT unique_id="PRODXML:0003">
      <PRINTER_NAME>Dev1</PRINTER_NAME>
      <COPIES>2</COPIES>
      <INCLUDE_INC>xm1.inc</INCLUDE_INC>
      <TO_NAME>Bill Smith</TO_NAME>
      <TO_COMPANY>Acme, Inc.</TO_COMPANY>
      <TO_CONTACTNUM>555-6545</TO_CONTACTNUM>
      <COVERSHEET>auto.csv</COVERSHEET>
    </PRINT>
  </DESTINATIONS>
</XML_FAX_SUBMIT>
```

```

<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">
This is a line of text for the cover sheet.
Here is another line of cover text.
</COVER_TEXT>
<BODY type="TXT" encoding="NONE" tm="1" lm="1" bm="1" font_name="Arial"
font_size="10" font_leading="12" font_pitch="12">
Info for body of the fax
&lt; &gt;
More More More
</BODY>
<ATTACHMENT>
<DATA type="RTF" encoding="NONE">
\rtf1\ansi\ansicpg1252\deflang1033{\fonttbl{\f0\fswiss\fcharset0 Arial;}}
{\*\generator Msftedit 5.41.21.2500;}\viewkind4\uc1\pard\f0\fs20 Put this text into the
document as an attachment.\par}
</DATA>
</ATTACHMENT>

<!-- ( this attachment is commented out ) -->

Doc files are done the same way except that they must be BASE64 encoded using
Encode64.exe first.

<ATTACHMENT>
<DATA type="DOC" encoding="BASE64">
OM8R4KGxGuAAAAAAAAAAAAAAAPgADAP7/CQAGAAAAAAAABAAAAVgAAAAAAA
EAAAIAAAAEEAAD+///AAAAAFUAAAD/////////////////////////////////
///////////////////////////////////////////////////////////
///////////////////////////////////////////////////////////
///////////////////////////////////////////////////////////
///////////////////////////////////////////////////////////
///////////////////////////////////////////////////////////
///////////////////////////////////////////////////////////
///////////////////////////////////////////////////////////
///////////////////////////////////////////////////////////
pcEATSAJBAA8BK/AAAAAAAEEAAAAAABAAUTIAAA4AYmpiauI94j0AAAAAAA
AAAABBYAL1oAAIBXAACAVwAAUS4AAAAAAA
( ... Cut many pages of BASE64 encoded data ... )

aW9uIG9mIGhvdyB0byBkbyB0aGlzLiB0b3R1IHRoYXQgdGhpcyBpcyBqdXN0IHRoZSBiYXNpY3Mu
DUNyZWF0ZSBibiBSRmF4U3VibW10IG9iamVjdC4NU2V0IHRoZSB0YXJnZXQgVVJMIGZvcib0aGUG
UmlnaHRGQVggc2VydMVyLCB1c2luZyBzZXRUYXJnZXRVUkwgbWV0aG9kLiBUaG1zIHRha2VzIGEg
</DATA>
</ATTACHMENT>

( comment ends ) -->

<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>

```

16.7.2 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>
```

16.7.3 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>
```

16.7.4 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></CUSTOM1>
</FAXSTATUS>
</XML_FAX_QUERY_REPLY>
```

16.7.5 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>
```

16.7.6 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>
```

16.7.7 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>
```

16.7.8 Testing the sample files

To test using the sample files

1. In Enterprise Fax Manager, stop the **Integration Module** service.
2. Open a command prompt.
3. Go to ..\OpenText Fax\Production\bin.
4. At the prompt type: “**parseXML**”.
5. Paste the path and file name of the **XML_FAX_SUBMIT_minimum.xml** file after “**parseXML**.”

You can do this by opening Windows Explorer onto the folder containing that file, then dragging and dropping the file onto the command line. The file’s default location is ..\OpenText Fax\Production\xml\examples\

6. Press **Enter**.

If the test is successful you can go to ..\OpenText Fax\Production\Inbox and find an .fcl file there. Inspect this file, and you see that the xml input has been transformed into fcl output. You may also do this with the **XML_FAX_SUBMIT.xml** sample file. If there is no file in the Inbox, double check that the Integration Module is stopped. If it is not, it will have picked up, processed, and deleted the .fcl file.

If the test is not successful an **error.fcl** file appears in the Inbox. Use it to troubleshoot.

Chapter 17

Creating FCL documents

To create documents with FCL, 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 commands in combination with the configured OpenText Fax Integration Module defaults 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.

17.1 The format of FCL commands

All FCL commands begin with two left braces ({{}) and end with two right braces (}}). Many of them also include arguments and variables, such as a delivery type or a fax number, within the braces.

Each FCL document must have at least one {{begin}} and one {{end}} command. The Integration Module will process all the data that appears between a {{begin}} and an {{end}} command as a discreet document. FCL documents may contain multiple {{begin}} and {{end}} commands, where the FCL between each set of commands will be rendered as a separate page. Data that does not appear between {{begin}} and {{end}} commands is ignored.

If the data stream from the host application will include double-byte characters, the file header of the FCL file must include the appropriate byte order marker (BOM).

For each document, you must also specify how the document will be processed. You can do this with either the {{fax}} command or the {{type}} command. The {{fax}} command must include either the fax number or an email address.

The FCL must also comply with any other OpenText Fax requirements as defined in your implementation of OpenText Fax. For example, if you require a billing code before sending a fax, the FCL must also include a billing code with either the {{billing}} or {{billing2}} commands.

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

```
 {{begin}}{{fax 503-555-1234}}
 {{attach ptc6mos.xls}}{{onerror fax 555-6892}}
 {{end}}
```

This example converts the spreadsheet ptc6mos.xls to a TIFF, then faxes it to 503-555-1234. If this fax does not go through successfully, the entire fax will be sent to 555-6892. All other functionality, such as margins and notifications, will use the values as set in the OpenText Fax EFM Integration Module Configuration window.

For a detailed list of available FCL commands, see “[FCL Commands](#)” on page 203.

17.2 Setting defaults for FCL documents

Use the OpenText Fax Integration Module Configuration program to set most default values. 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 [Reusing FCL Commands Across Many Documents](#).

17.2.1 Opening the Integration Configuration Program

To open the configuration program

1. On the Start menu, click **OpenText Fax**, and then click **OpenText Fax 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 **OpenText Fax Integration Module**.

17.2.2 Setting general defaults

To set general defaults

1. In the left pane of the **Integration Module Configuration** window, click **General**. The **General** settings appear in the right pane.
This setting can be overridden for a document with the {{rti}} command. For more information on this command, see “[FCL Commands](#)” on page 203.
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 {{csi}} command. For more information on this command, see “[FCL Commands](#)” on page 203.
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 {{rti}} command. For more information on this command, see “[FCL Commands](#)” on page 203.
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 “[Creating and attaching cover sheets and other files](#)” on page 61.
This setting can be overridden for a document with the {{type print}} FCL command. Enter a printer that is defined in Enterprise Fax Manager.
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 is defined in Enterprise Fax Manager.
This setting can be overridden for a document with the {{printer}} command. For more information on this command, see “[FCL Commands](#)” on page 203.

6. In the **Event Log Level** box, select at which level to log information in the Windows Event Log.
 - **None** No information.
 - **Terse** Critical errors only.
 - **Normal** Errors and major events only.
 - **Verbose** All significant events. Use temporarily to track and resolve problems.



Caution

Leaving the verbose log level for long periods of time can fill up the Event Log and may prevent new events from being logged.

7. In the **Server name** box, enter the name of the OpenText Fax server.
8. In the **Protocol** list, select the communication protocol for the OpenText Fax server.

17.2.3 Setting defaults for FCL processing

To set the default for FCL processing

1. In the left pane of the **Integration Module Configuration** window, click **FCL Processor**. The **FCL Processor** settings appear in the right pane.
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 specifies 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 “[FCL Commands](#)” on page 203.
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 OpenText Fax can attach up to 1024 pages to a fax.
6. By default, attached documents that are 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.

17.3 Sending basic FCL jobs

The following examples demonstrate how to use the FCL commands in combination to send single faxes, to broadcast faxes, and to use cover sheets.

The complete list of FCL commands is in alphabetic order in “[FCL Commands](#)” on page 203.

17.3.1 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 {{altnax}} command as a backup fax number if the primary number fails:

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

17.3.2 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. OpenText 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. “[Using cover sheets in a broadcast](#)” on page 167.

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.

- 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}}

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

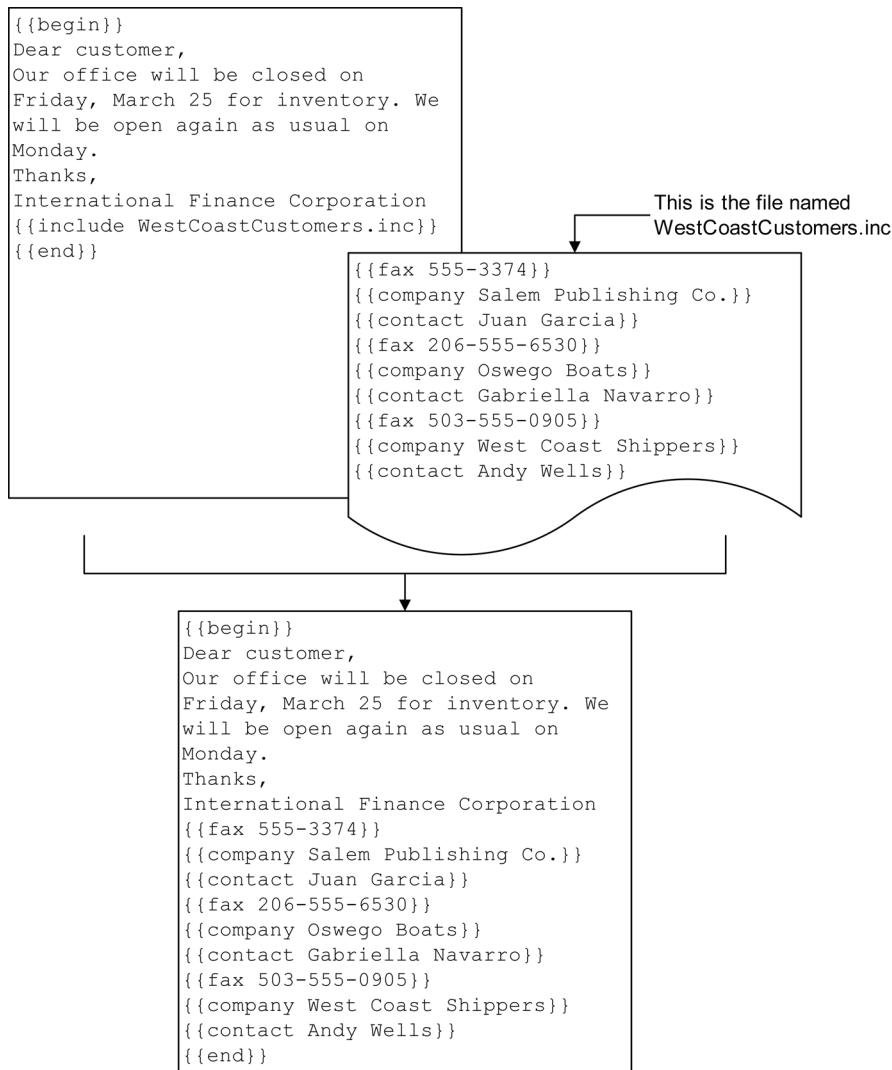
```
 {{begin}} Dear customer,
Our office will be closed on Friday, March 25 for
inventory. We will be open again as usual on Monday.
Thanks,
International Finance Corporation{{fax 555-3374}}
{{company Salem Publishing Co.}}
{{contact Juan Garcia}}
{{fax 206-555-6530}}
{{company Oswego Boats}}
{{contact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

17.3.3 Broadcast faxing 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.

Broadcast documents with an {{Include}} command

The following illustration shows the broadcast document with an {{include}} command that inserts multiple addresses, instead of the addresses themselves. The {{include}} command in the original data is replaced by the list of addresses found in the file WestCoastCustomers.inc.



By having multiple broadcast include files stored on the Integration Module, you can easily broadcast a document to whatever group of numbers you choose.

The following example sends the same fax to all fax numbers stored in the file called dealers.inc.

```

{{begin}}
{{include dealers.inc}}
{{form invoice.tif}}
Host data.
{{end}}

```

17.3.4 Creating a broadcast list

There are two ways to create an include file that contains a broadcast list:

- Create a list of FCL commands in any word processor, spreadsheet, or database program that can produce plain text, then manually copy the text file to the OpenText Fax\Production\Include folder. The text file must have the extension .inc or errors will occur.
- Create an FCL document that uses either {{file}} or {{list}} to create an include file, and then send the FCL document to the Integration Module. This method automatically stores the file in the OpenText Fax\Production\Include folder.



Note: Both {{file}} and {{list}} create and store files in the same folder. {{list}} removes all white space, blank lines, etc. from the input file, while {{file}} does not.

To create a file called dealers.inc in the OpenText Fax\Production\Include folder, send the following document to the Integration Module,

```
 {{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}}
```

The resulting file, Dealers.inc, contains 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.}}
```

17.3.5 Using cover sheets in a broadcast

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 *OpenText Fax - Administrator Guide (FXNET240400-AGD)*.

Table 17-1: Cover sheet keyword equivalents

.cov keyword	.pcl or .doc keyword	Notes
billing	billinfo1	15 characters maximum.
billing2	billinfo2	15 characters maximum.
comment	to_citystate tocitystate	59 characters maximum.

.cov keyword	.pcl or .doc keyword	Notes
company	to_company tocompany	59 characters maximum.
contact (fax)	to_name toname	59 characters maximum.
covertext	notetext	21 lines, each with each containing 69 characters maximum.
csi	from_phonenum fromphonenum	31 characters maximum.
email (email)	to_name toname	59 characters maximum.
fax	to_faxnum tofaxnum	31 characters maximum. Can contain alphabetical characters for macros.
onerror delete	faxflag_autodeleteall	Deletes successful documents.
onsuccess delete	faxflag_autodelete	
owner	from_name fromname	59 characters maximum.
priority	ucPriority	Can be low (0), medium or normal (1), or high (2) priority. The default is low.
unique_id	uniqueid	
voice	to_contactnum tocontactnum	31 characters maximum.
winsecid	userid username	The OpenText Fax 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.

17.3.6 Sending documents to a printer or file with FCL

To send documents 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}}
```

To send 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 OpenText Fax\Production folder.

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

The {{file}} and {{list}} commands create a plain text (ASCII) file with an .inc extension in the OpenText Fax\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 OpenText Fax\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 “[FCL Commands](#)” on page 203.

17.3.7 FCL Dashboard

The FCL Dashboard displays the failed FCL jobs. You can view the FCL Dashboard from the EFM or Web Admin user interface. You can perform any of the following actions from the dashboard:

Edit or correct the FCL job

1. Right-click on the failed FCL job.
2. Click **Edit**.
3. Update the job and click **Resubmit**.

Delete the failed FCL job

1. Right-click on the failed FCL job.
2. Click **Delete**.

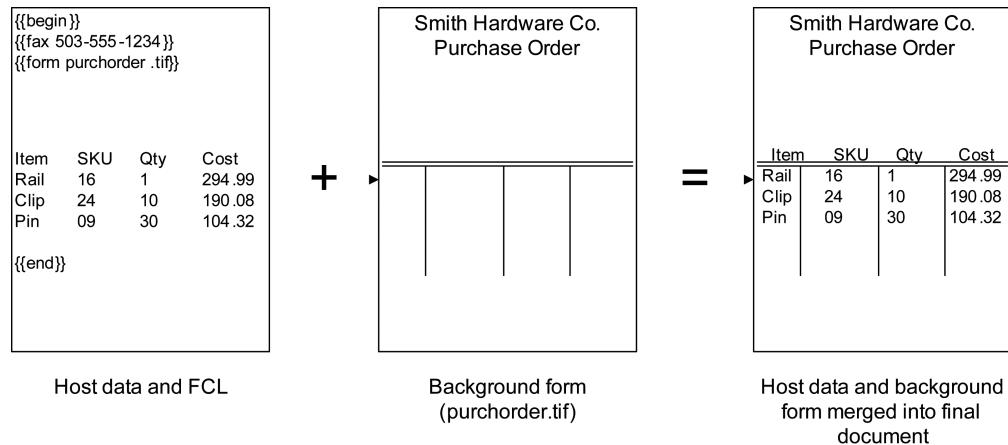
17.4 Creating and linking background forms

Background forms are 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.

17.4.1 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.



17.4.2 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. If you are duplicating an existing form, scan it at 200 dpi. If you are using a drawing package, you can use any graphic program that outputs a TIF file. In either case, all background forms must be in class F, group 3 or 4 TIF format and stored in the OpenText Fax\Production\Forms folder.

17.4.3 Linking background forms

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

In the following example the host data merges with a background form called `invoice.tif`, which resides in the OpenText Fax\Production\Forms folder.

```

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

```

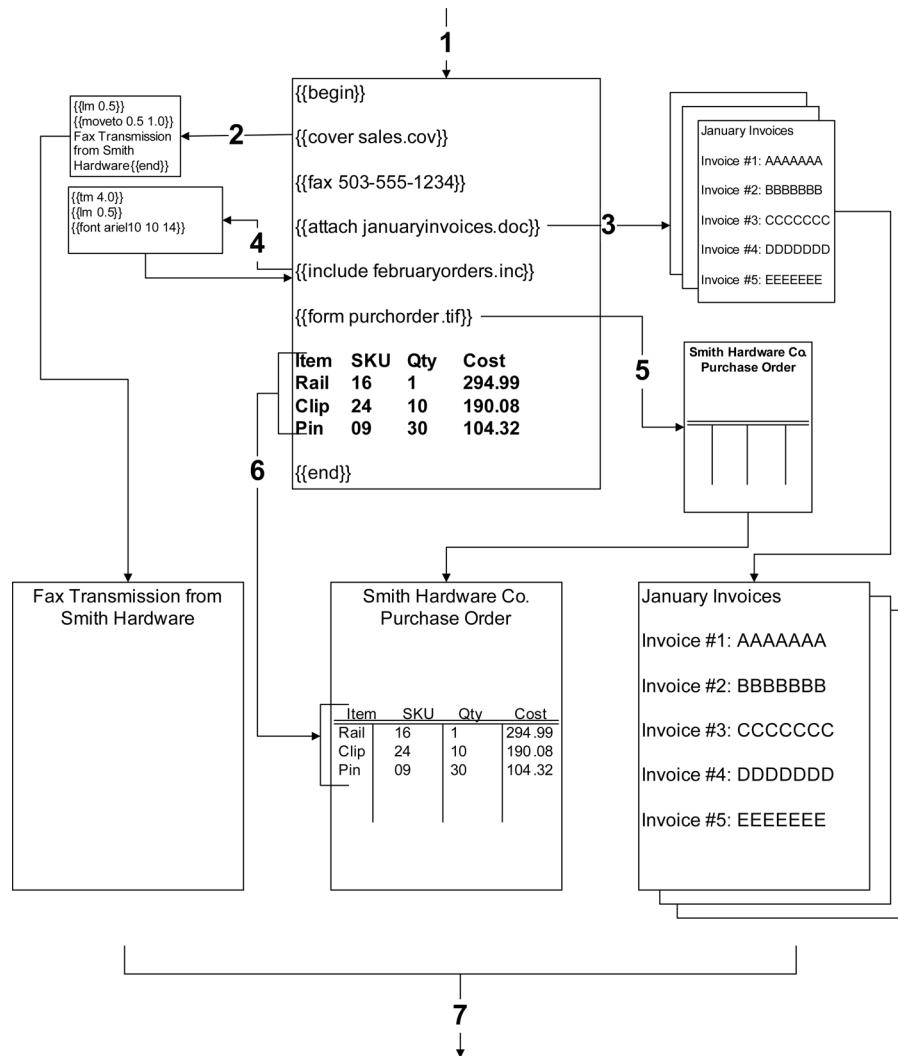
17.5 Example FCL documents

The following figure (Other Files Can Be Added to the Document) illustrates how other documents—such as cover sheets, background forms, and attachments—can be included in a document with the use of FCL commands.

1. The Integration Module receives the document data stream from the host. It contains the source document data and FCL commands. In the example, 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.csv` 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 had been printed on a pre-printed form.
7. The cover sheet, document, and attachment are faxed to 503-555-1234 in one transmission.

Other files can be added to the document



The following figure (FCL Document for an Invoice) illustrates the FCL-encoded document for an invoice.

1. The {{form}} command specifies that the file `Invoice.tif` will be added as the background for the fax. The figure Finished Invoice 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.

FCL document for an invoice

```

{{begin}}
{{faf 503 555 9182}}
{{form Invoice.tif}}
{{attach ptc6mos.xls}}
{{tm 1.25}}
{{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}}

```

Finished invoice

ptc6mos.xls	Data	Data	Data	Data	Data	Data	Data	Page 1 of 4
NW Distributors								
Invoice #				Date				
18277				8/8/01				
Sold to		Ship to						
Anne Paige Portland Trading Co. 11990 S.W. Capitol Hill Road Portland, OR 97034		Portland Trading Co. 11990 S.W. Capitol Hill Road Portland, OR 97034 (503) 555-9182						
55222 Your order # 8/4/01 Date ordered		13579 Our order # 8/7/01 Date shipped		John Bardolph Sales person 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 \$1,025.00		Sales tax \$75.00	Shipping \$75.00	Total \$1,175.00				

Chapter 18

Programming for the OpenText Fax API for Java

The OpenText Fax API for Java converts Java to XML on the host computer and then transmits it to the OpenText Fax Integration Module. The OpenText Fax XML Interface converts the XML to FCL. The OpenText Fax Integration Module can then process and send the document from the OpenText Fax server.

The OpenText Fax API for Java can submit an outbound document, query the OpenText Fax server for the status of a document, and perform actions (forward, delete, or create a library document) on previously sent documents.

The OpenText Fax API for Java requires the Sun Microsystems, Inc. Java Developer's Kit version 1.1.8 or later.

The API for Java provides access to the following functions and methods of transport in the OpenText Fax XML Interface.

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 specifics about the OpenText Fax API for Java classes, fields, and methods, see the OpenText Fax html Help files that are installed with the API in the folder `OpenText Fax\Production\XML\Java\Docs`.



Note: With OpenText Fax 10.6 FP1, the API for Java was changed to properly escape the following XML entities: single quote ('), double quote ("), greater than (>), less than (<), and ampersand (&). Any custom code created to escape these entities in previous versions must be changed to not escape them if the code is to be recompiled with the current API.

18.1 Installing the API for Java

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

18.1.1 Minimum system requirements

In addition to the minimum system requirements for the OpenText Fax server and OpenText Fax Integration Module, Microsoft Internet Information Server (IIS) version 7.0 or later must be installed on the OpenText Fax server.

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.

18.1.2 Installing the Java Interface

Follow the instructions for installing the OpenText Fax server in the *OpenText Fax - Installation guide (FXNET240400-IGD)* and use the following specific steps:

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

Complete the remaining instructions in the *OpenText Fax - Installation guide (FXNET240400-IGD)* for installing the OpenText Fax server. After the OpenText Fax server and Java/XML API component are installed and activated, configure an SMTP host and an IIS user account.

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

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

Additional Java development tools are located in the `Program Files\OpenText Fax\Production\XML\Java` folder.

18.1.3 Configuring the Java development computer to use the OpenText Fax Java interface

Copy `RFJavaInt.zip` from the `Program Files\OpenText Fax\Production\XML\Java` folder on the OpenText Fax 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, copy this file to the same location.

When the file has been copied, you must modify the `CLASSPATH` environment variable to include this `.zip` file.

If the `RFJavaInt.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\RFJavaInt.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.*;
```

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.

18.2 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 instructions and the following sample code for creating an outbound document include code that conforms to an XSD schema so you can create content using MSXML 6.0. However, implementing the XSD schema can break existing XML/Java integrations.

You can call the `submitEx` method in place of `submit`. This returns a string containing the unparsed XML returned by OpenText Fax. The XML contains the status of each recipient in the form of the example shown under [Sample code for creating an outbound document](#).

To send an outbound document

1. Create an `RFaxSubmit` object.
2. Set the target URL for the OpenText Fax server, using the `setTargetURL` method. This takes a `String`, NOT a `URL` object (it will create the `URL` object).

3. Use the **RFDoc** object **m_FaxDocument** in the **RFaxSubmit** object to set all the non-attachment information about the document.
 - Call **m_FaxDocument.setStyleSheet** to set the style sheet for the document.
 - Call **m_FaxDocument.setXMLNS** to specify the XML namespace.
 - Call **m_FaxDocument.setSenderInfo (...)** to set the information about the person sending the document. This method is 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 do not 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 is 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 ("").
 - 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.
5. When all the data is set, call the method. You get back a vector of RFStatus objects. Each object will contain the status for one recipient, stating whether it was passed on for sending or if there was an error.

18.2.1 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 an outbound fax object
    RFaxSubmit obFS = new RFaxSubmit();
    //Set the URL of the OpenText Fax server
    obFS.setTargetURL ("http://www.company.com/");
    obFS.m_FaxDocument.setStyleSheet ("XML_FAX_SUBMIT.xsl");
    obFS.m_FaxDocument.setXMLNS ("x-schema:C:\\Program Files (x86)\\RightFax\\Production\\xml
    \\schemas\\XML_FAX_SUBMIT_schema.xml");
    //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", "", "",
        "", "", "", "", "photoimage");
        } catch (RFNoFaxNumberException nfne) {
System.out.println (nfne.toString());
        } catch (RFInvalidIDException iide) {
System.out.println (iide.toString());
        }
        //Add an email 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());
        }
    }
}

```

18.3 Querying documents

To perform a query on the status of a document (or group of documents), you must set the criteria for each query.

To query documents

1. Create an RFaxQuery object.
2. Set the target URL for the OpenText Fax 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 is 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 CALENDAR 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).

18.3.1 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 OpenText Fax 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());
    }
}
```

18.4 Performing actions on documents

To perform an action on a document, you must have its unique ID. Using the unique ID, you can delete the document, forward it to another recipient, or use it to create a library document. To find a document's unique ID, perform a query for the document. See [Querying documents](#).

When you submit a document, you can create a unique ID for that document or let the OpenText Fax 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 OpenText Fax server.
- The last eight characters are a number unique to a document.

To perform an action on documents

1. Create an *RFaxAction* object.
2. Set the target URL for the OpenText Fax 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 is 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 the action was successful.

18.4.1 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 OpenText Fax 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());
    }
}
```

18.5 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 OpenText Fax server and the XML returned from the OpenText Fax server.

To turn on debug mode, call the `setDebug` method, passing it `true`.

18.6 Error and status codes for XML- and Java-based documents

The OpenText Fax API for Java returns 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 is 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.

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 OpenText Fax 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.
Submit		

Code	Message	Explanation
-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.
0-n	N/A	If you receive a message not listed in this table, it is a code universal to all OpenText Fax software. These are listed in the <i>OpenText Fax - Administrator Guide (FXNET240400-AGD)</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

Code	Message	Explanation
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>OpenText Fax - Administrator Guide</i> (FXNET240400-AGD)

Chapter 19

Creating FCL documents with InternetLink commands

The OpenText Fax InternetLink Module is a component of the OpenText Fax 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.

19.1 System requirements

The InternetLink Module requires the following:

- OpenText Fax Enterprise Server and Integration Module already installed
- Network connection to the Internet
- SMTP gateway on the network
- TCP/IP connection from the OpenText Fax Integration Module to the network

19.2 Verifying the name of the email server

Before you begin sending documents with the InternetLink Module, verify the name of your email server.

To verify the name of your email server

1. On the OpenText Fax server, open EFM.
2. In the left pane, expand the server, and then click **Services**.
3. In the list of services in the right pane, double-click **OpenText Fax Server Module**.
4. On 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.

19.3 InternetLink FCL commands

Documents sent via the InternetLink Module require six FCL commands. Without these commands, the document will not be sent.

Required FCL Commands

Command	Description
<code>{{begin}}</code>	Indicates the beginning of a document. The Integration Module will process all the data that appears between a <code>{{begin}}</code> and <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. This command must appear as the first command in each InternetLink document.
<code>{{end}}</code>	Indicates the end of a document. The Integration Module will process all the data that appears between a <code>{{begin}}</code> and <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. This command must appear as the last command in each InternetLink document.
<code>{{from}}</code>	Sender's email address.
<code>{{subject}}</code>	Subject line of the email message.
<code>{{to}}</code>	Recipient's email address.
<code>{{type email}}</code> or <code>{{type mime}}</code>	<code>{{type email}}</code> sends a document as the editable body of an email message. <code>{{type mime}}</code> sends a document as an attachment to an empty email message.

The commands `{{begin}}` and `{{type}}` can be replaced with a "shortcut" command, `{{begin mime}}` or `{{begin email}}`.

19.3.1 Choosing document types

The document type determines the type of document that the recipient receives via email. The type of document is specified with the {{type}} command.

- {{type email}} converts the document to the editable body text of an email 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 email 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.

19.3.2 Choosing image types

The image type refers to the type of graphic file that is created when you use the {{type mime}} command, PDF, searchable PDF (SPDF), TIF, or PCX. To specify an image type, use the {{imagetype}} command (see “[IMAGETYPE](#)” on page 225).

The default graphic format for {{type mime}} documents is Group 4 TIFF.

19.3.3 Using include files with the InternetLink Module

An empty include file called `Mime.inc` is created in the `OpenText Fax\Production\Include` folder. You can insert plain text (but not FCL, formatted text, or graphics) in `Mime.inc`. The 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.

19.4 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 “[FCL Commands](#)” on page 203.

In the following example, `Program.xls` is attached in its native format (as a Microsoft Excel document) to the host document:

```
 {{attach c:\\IST Files\\Programs.xls native}}
```

If you create a {{type mime}} document that contains the {{attach}} command with the native option, the attached document becomes the second attachment to an empty email message.

If you use the {{attach}} command without the native option, the attached document and the host document are merged into one graphic file that is attached to an empty email message. You determine the image type of this graphic file with the {{imagetype}} command. See [Choosing image types](#).

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.

19.4.1 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, see [Using include files with the InternetLink Module](#).

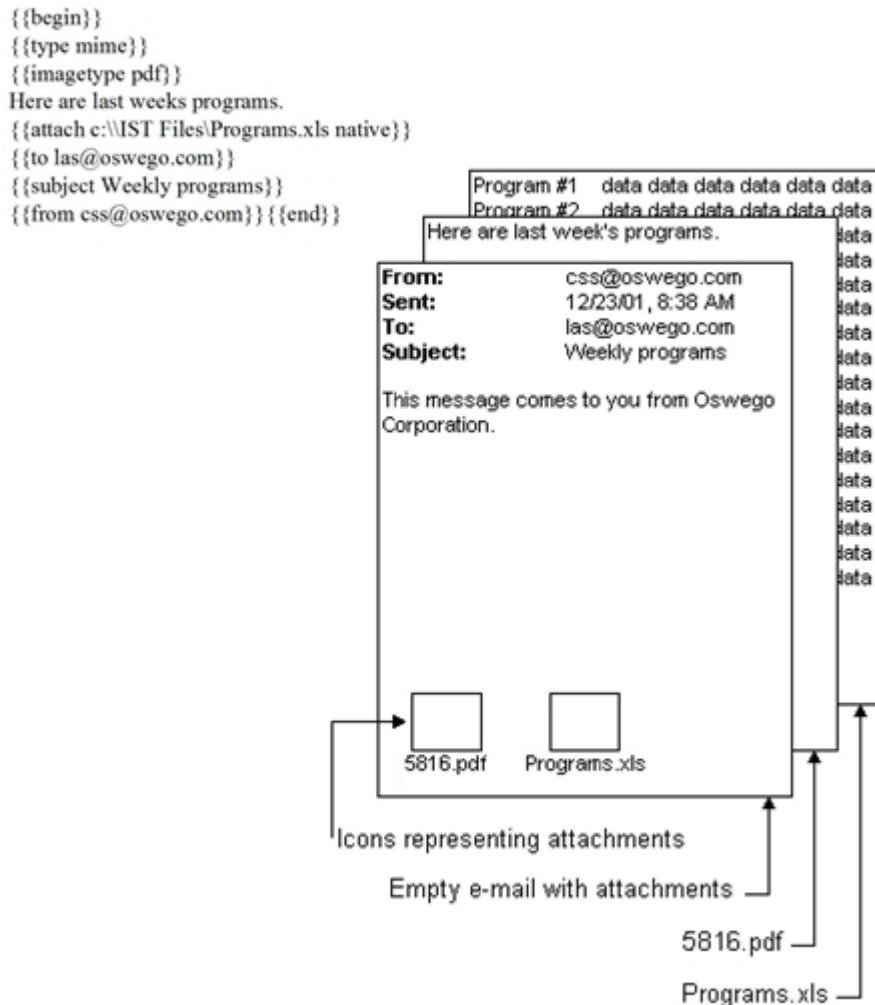


Figure 19-1: Attaching a file in its native format

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 5816.pdf.

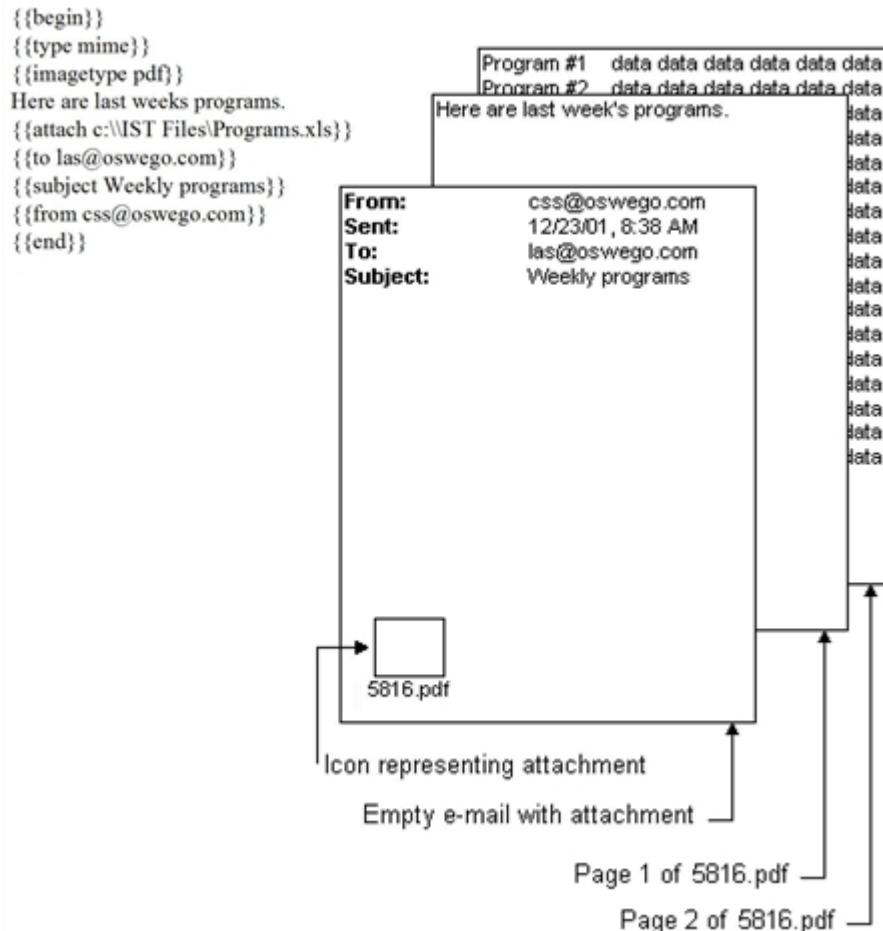


Figure 19-2: Attaching a file converted to PDF format

19.5 Sending documents as email if faxes fail

With the InternetLink Module, you can send email 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 email address}}}
```

Unlike the {{type}} command, the email 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 email 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, it will be sent as a PDF file attached to an empty email 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 email, all email addressing options (such as {{to}} and {{cc}}) in the original document are replaced with the email 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.

19.6 Receiving notification when a fax fails and is sent as an email

To create a notification that a fax has failed and has been sent as an email, include the ^type^ keyword in the notification template. See "[Creating notification messages with FCL](#)" on page 93

Chapter 20

OpenText OpenText Fax Connector for Oracle Purchasing 11i

With the OpenText Fax Connector for Oracle Purchasing 11i, you can send purchase orders, releases, and change orders from the Oracle Purchasing 11i application, a component of the Oracle E-Business Suite, via your OpenText Fax server.

20.1 Activating the Oracle Purchasing 11i Connector

The files required by the Connector for Oracle Purchasing 11i are installed on all OpenText Fax servers during the server installation. Enable the OpenText Fax Integration Module through licensing and activation to use it with Oracle.

For information on activating new components on the OpenText Fax server, see *OpenText Fax - Installation guide (FXNET240400-IGD)*.

20.2 How the Oracle Purchasing 11i Connector works

The Connector for Oracle Purchasing 11i fax- and -email enables Oracle Purchasing 11i by intercepting data from the print stream sent from Oracle. The connector affixes elements of the OpenText Fax Facsimile Command Language (FCL) to the document data, checks FCL syntax, and then faxes or emails the document.

20.2.1 Sending documents from Oracle 11i

The Oracle Purchasing 11i user interface contains a **Fax Number** text box where a user can type a recipient fax number. The user can also enter an email address in this text box. The Oracle Purchasing 11i Connector uses this fax number or email address to send the document to a recipient.

20.2.2 Document templates

The OpenText Fax Connector for Oracle Purchasing 11i includes customizable Integration Module templates that have been created for Oracle Purchasing 11i:

- purchase order
- release
- change order

These templates define the layout of each type of document, including information such as text margins and fonts.

You can customize the default document templates by:

- Placing graphics, such as logos, on the template.
- Including background forms, such as a purchase order form.
- Specifying a cover sheet to send with each document.
- Setting margins and fonts, choosing a printer, and changing the page orientation.
- Configuring documents to be printed, faxed, or emailed.
- Specifying a notification method and notification templates.
- Adding FCL to the template.
- Re-naming the templates.

The following is an outline of the data flow process between the Oracle and OpenText Fax servers

1. Print output is sent from Oracle Purchasing 11i to the OpenText Fax server.
2. OpenText Fax detects the {{doctype}} FCL command and accesses the relevant document template.
3. FCL in the document template is added to original Oracle document.
4. OpenText Fax reformats and transmits the document based on FCL-defined parameters in the document template.

20.3 Editing the default templates

The default settings in the templates are the default values established in the Integration Module. When you edit the Oracle Purchasing 11i templates, the settings you make override the defaults. For more information about defining the appearance of documents, see “[Defining fax page appearance](#)” on page [75](#).

Do not create new templates. Only the default templates for a change order, a purchase order, and a release are supported.

Although you can edit the default document templates, do not delete them.

To edit a document template

1. On the OpenText Fax server in Windows Explorer, go to OpenText Fax\Production\Bin and run ERPConnector.exe.
This opens the **Oracle Purchasing 11i Connector** window.
2. To re-name a template in the **Template Name** list, select the name, click **Rename**, and enter the new name. This changes the name that appears in the display only. It does not change the file name.
3. In the **Template Name** list, select the template to edit.
4. Click **Edit**. The edit dialog box appears in the right pane of the window.
The editing options are described on the following pages:

- “Adding a graphic image or a background form” on page 197.
 - “Adding a cover sheet” on page 197.
 - “Configuring the layout” on page 198.
 - “Faxing a copy of a document that was transmitted” on page 198.
 - “Faxing a copy of a document that cannot be transmitted” on page 199.
 - “Specifying the notification method” on page 199.
 - “Adding FCL code to the template” on page 200.
5. After making your changes, click **Save** on the **File** menu, or click the **Save** icon on the toolbar.

20.3.1 Adding a graphic image or a background form

To add a graphic

1. Click the **Forms** tab.
2. In the **Form File** box, enter the path to the background form to include with the document.
3. To specify the position of the upper-left corner of the background form, select the **Place at Position** check box. The default is 0,0 from the top-right corner of the document.
4. In the **Horizontal Shift** and **Vertical Shift** boxes, enter the x- and y-coordinates for the upper-left corner of the background form.
5. In the **Logo File** box, enter the path to a graphic to appear in the document.
6. To specify the position of the upper-left corner of the logo, select the **Place at Position** check box. The default is 0,0.
7. In the **Horizontal Shift** and **Vertical Shift** boxes, enter the x- and y-coordinates for the upper-left corner of the graphic.

20.3.2 Adding a cover sheet

To add a cover sheet

1. Click the **Cover Page** tab.
2. In the **Cover File** box, enter the path to the cover sheet file to include with the fax.

20.3.3 Configuring the layout

To configure the layout

1. Click the **Layout** tab.
2. In the **Name** list, click the font name.
3. In the **Points of Leading** box, enter the value for the vertical spacing of the font.
4. In the **Pitch** box, enter the value (in 1000ths of an inch) for the horizontal spacing of the font.
5. In the **Top**, **Left**, and **Bottom** boxes, enter the width of the page margins.
6. Click **Default**, **Portrait**, or **Landscape** to specify the orientation of the page.
7. In the **RightFax Printer** box, enter the name of the a printer defined in Enterprise Fax Manager. This creates a default printer for documents that use the document template that you are creating.

20.3.4 Faxing a copy of a document that was transmitted

To fax a copy

1. Click the **On Success** tab.
2. Select one of the following options.
 - **Default.** Performs the default action specified in the Integration Module Configuration program.
 - **Nothing.** No action is taken.
 - **Fax.** Fax the document to another recipient.
 - To send the fax to the default number specified in the Integration Module Configuration program, click **Fax to Default Fax Number**.
 - To specify the fax number, click **Fax to Number**. Enter the fax number. If you are using the Enterprise Integration Module, you can enter an email address.
 - To send only the first page of the document, select the **Fax First Page Only** check box.
 - To delete the fax image from the fax server after transmission, select the **Delete Fax when Finished** check box.

20.3.5 Faxing a copy of a document that cannot be transmitted

To fax a copy of a document

1. Click the **On Error** tab.
2. Select one of the following options.
 - **Default.** Performs the default action specified in the Integration Module Configuration program.
 - **Nothing.** No action is taken.
 - **Fax.** Fax the document to another recipient.
 - To send the fax to the default number specified in the Integration Module Configuration program, click **Fax to Default Fax Number**.
 - To specify the fax number, click **Fax to Number**. Enter the fax number. If you are using the Enterprise Integration Module, you can enter an email address.
 - To send only the first page of the document, select the **Fax First Page Only** check box.
 - To delete the fax image from the fax server after transmission, select the **Delete Fax when Finished** check box.

20.3.6 Specifying the notification method

Methods for sending notification messages to a host application are configured in the Integration Module. For information and instructions, see “[Document transmission and notifications](#)” on page 91.

Notification messages, including templates, are determined by the {{notifyhost}} FCL code. For more information on notification messages, see “[Creating notification messages with FCL](#)” on page 93.

To specify a notification method

1. Click the **Notify** tab.
2. In the **Notification Method** list, select a notification option. The options in this list are created in the Integration Module Configuration program.
3. In the **Successful Template** list, select the template for notification messages about successful transmissions.
4. In the **Unsuccessful Template** list, select the template for notification messages about failed transmission attempts.

20.3.7 Adding FCL code to the template

You can extend the template by adding FCL code. The complete list of FCL codes, their syntax, and the placement of codes within documents is provided in “[FCL Commands](#)” on page 203.

To add an FCL code

1. Click the **Custom FCL** tab.
2. In the **FCL commands to be inserted before document text** box, type the FCL codes.
3. In the **FCL commands to be inserted after document text** box, enter FCL codes.

20.4 Sending documents by email

The user can enter an email address in the Oracle 11i **Fax Number** box. When a document is sent as email, it is attached as a TIFF image to an empty email message.

The subject line of the email message is “E-Document,” and the sender’s email address is `admin@<tcp/ip domain>`, where `<tcp/ip domain>` is the following value in the Windows Registry:

`HKLM\System\CurrentControlSet\Services\Tcpip\Parameters\Domain`

or

`HKLM\System\CurrentControlSet\Services\Tcpip\Parameters\DhcpDomain`

To notify an administrator or the sender when an email address is invalid, you have the following options:

- Send a message that notifies the sender that the email address is invalid.
- Print or fax the document to another recipient.

Chapter 21

OpenText Fax Integration Module programs

The following table lists the programs that are installed with the OpenText Fax Integration Module in the `OpenText Fax \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.

Program	Description	Command line syntax
<code>Bufdir.exe</code>	Reads a directory and executes a command on each file that is placed in the directory.	<code>bufdir [options] directory</code>
<code>Capture.exe</code>	Reads input from a serial port at different rates and in different modes and executes a command on data that is received.	<code>capture [options] input</code>
<code>Dbnotify.exe</code>	Updates databases from a Win32 command line.	<code>dbnotify [options] Data Source Name</code>
<code>Diagtiff.exe</code>	Diagnoses TIFF image files and provides the contents of their tags.	<code>diagtiff [options] infile [infile ...]</code>
<code>Encode64.exe</code>	Encodes files using base64 encoding.	<code>encode64 [options] infile outfile</code>
<code>Fax2mapi.exe</code>	Sends MAPI32 messages from the Win32 command line.	<code>fax2mapi [options]</code>
<code>Fax2note.exe</code>	Sends messages to Lotus Notes.	<code>fax2note [options]</code>
<code>Hd.exe</code>	Displays files in hexadecimal format.	<code>hd [options] input file</code>
<code>Hf.exe</code>	Sends data to a TCP/IP port.	<code>hf [options]</code>
<code>Hlpisend.exe</code>	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.	<code>hlpisend [options]</code>

Program	Description	Command line syntax
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] filename
Multitif.exe	Breaks one multiple-page TIFF image into a document consisting of single-page TIFF images.	multitif [options] documentnumberstartingpage
Nplisten.exe	Creates a named pipe and executes a command on data received via the named pipe.	nplisten [options]
Portlstdn.exe	Executes a command on a data stream received via a TCP/IP port.	portlstdn [options] command
Pssplit.exe	Splits a PostScript file to create separate pages.	pssplit [options] infile outputdirectory
Tee.exe	Transcribes standard input to standard output and makes copies in the file.	tee [options] outfile
Tiffbind.exe	Combines single-page TIFF images into one multiple-page TIFF image.	tiffbind [options] outputtiff inputtiff {input tiff ...}
Tog3.exe	Converts differing file formats of black and white TIFF images to Group 3, Group 4, PCX, and GIF images.	tog3 [options] infile outfile
Uuencode	Encodes files using uuencode.	uuencode [options] infileoutfile

Chapter 22

FCL Commands

This section 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.

22.1 ABORT

Causes the software to ignore a document before it is faxed. You can place this command anywhere in the FCL document.

Syntax
`{{abort}}`

22.2 ADDCOPIES

Increments the number of copies to print. This affects documents with the {{type print}} command.

Syntax
`{{addcopiesnumber}}`

Example:

```
{{addcopies 3}}
```

The example increments the number of printed copies of the current document by three.

See also “[TYPE](#)” on page 252.

22.3 ALTFAX

Stores an alternate fax number for document if the primary fax number fails.

The {{altnumber}} 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 {{altnumber}} command after every new {{fax}} or {{execute}} command.

Syntax
`{{altnumberfax number}}`

Example:

```
{{altnumber 503-555-5458}}
```

The example sets the alternate fax number for the document to (503) 555-5458.

22.4 APPROVAL

Holds the document for approval in a FaxUtil mailbox.

Syntax

`{{approval}}`

Example:

```
 {{approval}}
```

22.5 ATTACH

Ends the current page (unless it is blank) and attaches one or more files. It may also delete 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 OpenText Fax\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.

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]}}`

```
 {{attach PoInv.doc delete}}
 {{attach "C:\Program Files\OpenText Fax\Production\MyDocs.doc" native}}
```

The first example converts `PoInv.doc` to fax format (`.tif`) and attaches it to the document. After the fax is generated, `PoInv.doc` is deleted. Because no path is specified, `PoInv.doc` is located in the folder `OpenText Fax\Production\Forms`.

The second example can be used with the InternetLink Module to send the document via email. The base document would be sent as an email, and `PoInv.doc` would be attached to that email as a native file: it would be left as a Microsoft Word (`.doc`) file and not converted to fax format.

22.6 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*

{{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](#), [DELAY](#), and [TIME](#).

22.7 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.

If you licenced the InternetLink Module, you can use the {{begin}} command to specify either mime or email format when sending documents by email. Email format sends the contents as an editable body text of the email message. Mime format sends the content as an uneditable graphic attachment to an empty email message. If you place the email type in the {{begin}} command, you do not need to use the {{type}} command.

Syntax

{{begin*emailtype*

{{begin}}
{{begin email}}
{{begin mime}}

See also [END](#).

22.8 BEGINCVT

This command embeds a document in an FCL document. The fax server converts the document to a TIF image. This command must be used with the command {{endcvt}}.

You can also encode the embedded document using BASE64 (for any type of binary files) or QUOTEDPRINTABLE (for files that use only the human-readable character set). If no encoding sequence is specified, the data is assumed to be 8-bit binary.

Syntax

```
 {{begincvt} fileName [base64|QuotedPrintable]}}
```

```
 {{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](#).

22.9 BEGINNATIVE

{{Beginnative}} is part of a set of commands that insert data into an email message.

Data entered between the {{beginnative}} command and the {{endnative}} command is processed into an email message. You may specify a filename, body text, format, encoding, and other parameters as described in the following table.

Parameter	Description
"filename"	<p>File name for the data. You need not supply a path with this file name because the data is stored between the {{beginnative}} and {{endnative}} commands.</p> <p><code>{{beginnative "File.pdf" inline mediatype=application/pdf}}</code> Here are the files you wanted <code>{{endnative}}</code></p> <p>In this example, OpenText Fax will suggest to the recipient email client that "Here are the files you wanted" be displayed as an embedded object in the body of the email message. If the recipient email client cannot perform this task, then an icon labeled <code>File.pdf</code> will appear in the email message. The content of <code>File.pdf</code> will be "Here are the files you wanted."</p>

Parameter	Description
body	<p>This sends the data that appears between the {{beginnative}} and {{endnative}} commands as text in the body of the email message.</p> <p>Using the body option, you can specify alternative file formats when you want to give the recipient's email software a choice of how to display the message.</p> <p>For example, you can send a text file and an HTML file. Generally, the recipient's email software will display the file in the format that it supports. It will not display both files.</p> <p>The sequence of the file formats determines in which format the recipient's email software will display the message. 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 show text first.</p> <p>To specify alternative formats, use multiple {{beginnative}} commands with the body option.</p> <p>To send a document that contains both HTML and text, you might use the following FCL:</p> <pre data-bbox="869 1178 1323 1619">{{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>Drew,
 Here are the files that you asked for. </BODY></HTML> {{endnative}} {{end}}</pre>

Parameter	Description
inline	<p>This suggests to the recipient email client that the data between {{beginnativ}} and {{endnativ}} commands be displayed as an embedded object in the body of the email message. If the recipient email 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 email message. The icon can be opened with the appropriate application.</p>
mediatype=	<p>Specifies the nature of the data between {{beginnativ}} and {{endnativ}} 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.</p> <p>text/plain</p> <p>The top-level media type identifies a general type of data, and the subtype identifies a specific format for that type of data. A media type of image/tif is enough to tell a recipient email client that the data is an image, even if the recipient email client cannot interpret the specific image format of .tif.</p> <p>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):</p> <ul style="list-style-type: none"> application/ms-word application/pdf audio/mid audio/wav image/gif text/html text/xml video/mpeg <p>If you do not supply a media type, the document most likely will not fail because most current email clients can detect the media type.</p>

Parameter	Description
encoding	You can specify either BASE64 or QUOTEDPRINTABLE for encoding the attachment. The data between {{beginnative}} and {{endnative}} commands will be encoded using the specified scheme. If these options are not specified, then the data will not be encoded (8 bit).

{{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 email message.

The inline option suggests to the recipient email client that the data between {{beginnative}} and {{endnative}} commands be displayed as an embedded object in the body of the email message.

If the recipient email 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 email message. The icon can be opened with the appropriate application.

{{Beginnative}} requires {{endnative}}.

Syntax

{{beginnative "filename"}[body][inline][mediatype=type] [encoding]}

```
 {{beginnative "Body.txt" body mediatype=text/plain}}
```

See also

[ENDNATIVE](#)

22.10 BILLING

Specifies the billing code of the fax owner.

Syntax

`{{billingcode}}`

```
{billing 1234}
```

The example specifies that the billing code associated with this fax is 1234.

See also

[“BILLING2” on page 211](#)

`{{Billing}}` translates to the OpenText Fax server field billinfo1. See [“Using cover sheets in a broadcast” on page 167](#)

22.11 BILLING2

Specifies a secondary billing code of the fax owner.

Syntax

`{{billing2code}}`

```
{ {billing2 4567}}
```

The example specifies that the billing code associated with this fax is 4567.

See also

[“BILLING” on page 211](#)

`{{billing2}}` translates to the OpenText Fax server field billinfo2. See [“Using cover sheets in a broadcast” on page 167](#)

22.12 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

`{{bmmargin}}`

```
{ {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 228](#)

[“TM” on page 251](#)

[“UNITS” on page 256](#)

22.13 BOX

Draws a box in the current line width at the specified coordinates. It can also fill 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 must be 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"]}}`

```
{ {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 222](#)
[“LINEWIDTH” on page 227](#)
[“PLACEXY” on page 240](#)
[“RBOX” on page 242](#)
[“RFILLBOX” on page 244](#)
[“UNITS” on page 256](#)

22.14 CC

Stores the SMTP address for a carbon copy recipient of a document sent by email. To store multiple addresses, repeat this command for each recipient.

Syntax

`cc address`

```
{ {cc Justin@OKtires.com} }  
{ {cc Tommy@OKtires.com} }
```

The examples send a copy of the email message to both Justin@OKtires.com and Tommy@OKtires.com.

See also

[“TO” on page 252](#)

“FROM” on page 225
“SUBJECT” on page 250

22.15 CLEARTABS

Removes all tab stops from the document.

Syntax

{{cleartabs}}

{{cleartabs}}

See also

“SETTAB” on page 247
“TAB” on page 250

22.16 COMMENT

Stores a user-defined message specific to the document.

Syntax

{{comment *comments***}}**

{{comment Inv# 12345}}

The example associates the comment “Inv# 12345” with the document.

{{Comment}} translates to the OpenText Fax server field `to_citystate`. See “Using cover sheets in a broadcast” on page 167

22.17 COMMENTFORRECORDS

Stores a user-defined message that is associated with each of the faxes in a broadcast.

Syntax

{{commentforrecords *comments***}}**

{{commentforrecords MRN 5555555}}

The example associates the comment “MRN 5555555” with the faxes in a broadcast.

Comments in the **{{commentforrecords}}** command appear in FaxUtil and FaxUtil Web.

22.18 COMPANY

Stores a company name for the document.

Syntax

`{{companyname}}`

```
{ {company ABC Incorporated} }
```

The example associates the company name “ABC Incorporated” with the document.

`{{Company}}` translates to the OpenText Fax server field `to_company`. See “[Using cover sheets in a broadcast](#)” on page 167

22.19 CONTACT

Stores the contact name for the document.

Syntax

`{{contactname}}`

```
{ {contact Kim Boston} }
```

The example associates the contact name “Kim Boston” with the document.

`{{Contact}}` translates to the OpenText Fax server field `to_name`. See “[Using cover sheets in a broadcast](#)” on page 167

22.20 CONVERSION_BIAS

Overrides the default conversion bias set for the server.

Syntax

`{{conversion_biasbias}}`

```
{ {conversion_bias text} }
```

Enter the word or the number for the bias option:

- **none** or **1** to not optimize.
- **images** or **2** to optimize for graphical fidelity.
- **text** or **3** to optimize for legibility of text.
- **highcontrast** or **4** to optimize for both text and images.

22.21 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 OpenText Fax\Production\Covers (this path is for OpenText Fax Integration Module cover sheets; OpenText Fax server cover sheets are stored in a different folder on the OpenText Fax server). If the complete file name is not found, the OpenText Fax Integration Module tries to open the file by adding a .cov extension. If this fails, the cover sheet is assumed to be a OpenText Fax.pcl or .doc cover sheet. If no cover command is specified in a document, the default cover sheet defined in the OpenText Fax Integration Module Configuration program is used.

Syntax {{cover*template*}}

```
 {{cover sales}}
```

The example specifies the sales or Sales.cov file in OpenText Fax\Production\Covers should be used as a template for the cover sheet of the current document.

22.21.1 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 OpenText Fax Integration Module Configuration program.

Rfdefault

The default cover sheet action associated with the user is performed.

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 OpenText Fax server (.pcl) cover sheet is sent.  
 {{COVER PROD.COV}} - Specified OpenText Fax 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 OpenText Fax server (.pcl) cover sheet is sent.  
 {{COVER PROD.COV}} - Specified OpenText Fax Integration Module (.cov) cover sheet is sent.
```

22.22 COVERTEXT

This command creates text that appears on the cover sheet. Input between {{covertext}} and {{“ENDCOVERTEXT” on page 219}} will be displayed as a text block in the document cover sheet.

{{Covertext}} translates to the OpenText Fax server field notetext. See “[Using cover sheets in a broadcast](#)” on page 167

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.

0 (zero) is the only option for OpenText Fax server (.pcl or .doc) cover sheets. See “[Using cover sheets in a broadcast](#)” on page 167



Note: Do not embed PostScript commands in the text between {{covertext}} and {{endcovertext}}. They will not be stripped out.

Syntax {{covertext*indexnumber*}}

{{covertext 0}}

22.23 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

{{csistring}}

{{csi 503-555-5481}}

This example prints “503-555-5481” on the cover sheet.

{{Csi}} translates to the OpenText Fax server field from_phone.

See “[Using cover sheets in a broadcast](#)” on page 167

22.24 DATE

Sets the month, day, and year in which to send the current document (year is optional). {{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 OpenText Fax Integration Module uses the date the FCL input file was received from the host.

Syntax

{{date}}{month/day/year}}

```
  {{date 11/2}}
  {{date 11/20/2014}}
```

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, 2014.

See also

["TIME" on page 251](#)

22.25 DELAY

Schedules the document to be sent at a later time by the specified number of minutes.

Syntax

{{delay}}{minutes}}

```
  {{delay 10}}
```

The example adds 10 minutes to the scheduled send time of the document.

See also

["BATCH" on page 205](#)

["DATE" on page 217](#)

["TIME" on page 251](#)

22.26 DEPT

This string can be used to specify the department of the fax owner or the recipient's department.

Syntax
`{{dept
department}}`

```
 {{dept Sales}}
```

The example specifies that the department associated with this fax is "Sales".

22.27 EMAIL

Stores the sender's email 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
`{{emailaddress}}`

```
 {{email user@acme.com}}
```

The example stores the email address user@acme.com for use in notifications.

22.28 EMPID

Specifies the employee ID of the fax owner.

Syntax
`{{empidID}}`

```
 {{empid 555-111-2222}}
```

The example specifies that the employee ID associated with this fax is 555-111-2222.

22.29 END

Terminates the current document. 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 is rendered as a separate page.

Syntax
`{{end}}`

```
 {{end}}
```

See also

["BEGIN" on page 205](#)

22.30 ENDCOVERTEXT

Ends the storing of cover sheet text and tells the OpenText Fax Integration Module to again interpret incoming data.

Syntax

`{{endcovertext}}`

`{{endcovertext}}`

22.31 ENDCVT

Ends the conversion of an embedded foreign document that was begun with the `{{begincvt}}` command.

Syntax

`{{endcvt}}`

`{{endcvt}}`

See also

[“BEGINCVT” on page 206](#)

22.32 ENDNATIVE

Ends the processing of an attached native document that was begun with the `{{beginnative}}` command.

Syntax

`{{endnative}}`

`{{endnative}}`

See also

[“BEGINNATIVE” on page 206](#)

22.33 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 227](#)
["STARTPOLY" on page 249](#)

22.34 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 OpenText Fax WorkServer, and if the FCL document contains a print command, then the {{execute}} command will force printing to occur from the WorkServer, regardless of whether the Print service is enabled.

Syntax

`{{execute}}`

Example:

```
{{execute}}
```

See also

["END" on page 218](#)
["FAX" on page 220](#)

22.35 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 remain the same as in the original. To broadcast to many different destinations, include a fax command for each destination. If you have the InternetLink Module, then you can also insert an email address into the fax command and documents will be emailed.

Syntax

`{{faxnumber}}`

`{{fax emailnumber}}`

Example:

```
    {{fax 555-555-5458}}
```

The example assigns the fax number 555- 555-5458 to the document.

See also

["TYPE" on page 252](#)

`{{Fax}}` translates to the OpenText Fax server field **to_faxnum**. See “Using cover sheets in a broadcast” on page 167.

22.36 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}}
```

22.37 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 `OpenText Fax\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 `OpenText Fax\Production\Include\test.inc`.



Note: Newly-created include files may not be immediately available because of Windows delays. If this occurs, try placing the `{file}` command in a separate document, and submit it before the document that uses the `{include}` command to reference the file.

See also

“LIST” on page 228

22.38 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; the default is black.

Syntax

```
 {{fillbox coord1 coord2 color}}
```

Example:

```
 {{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 212](#)
[“LINEWIDTH” on page 227](#)
[“RBOX” on page 242](#)
[“RFILLBOX” on page 244](#)
[“UNITS” on page 256](#)

22.39 FONT

Changes the font currently in use. You can also change the attributes of the font.

The available font attributes are described in the following table.

Attribute	Description
Name	<p>The name of the font. This can be one of the fonts installed with the OpenText Fax Integration Module, a TrueType font, or any font supported by Microsoft Windows.</p> <p>Fonts are installed in OpenText Fax \ Production\Fonts. The font name is used for all subsequent text until another {{font}} command appears in the document or until the end of the document.</p>
Size	<p>The size of a Windows font in points. The default is 12. The size can range from 3 to 288.</p> <p>This attribute is not available for the installed fonts.</p>
Leading	<p>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.</p> <p>A leading of 12 points is 6 lines per inch (72 divided by 12 equals 6). You can enter leading in decimals.</p>
Pitch	<p>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.</p>
Weight	<p>The weight of a Windows font. Weights are thin, extralight, light, regular, medium, semibold, bold, extrabold, and heavy.</p> <p>This attribute is not available for the installed fonts.</p>
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 [“Selecting and configuring fonts” on page 86](#).

Syntax

`{{fontname [size=##] [leading=##] [pitch=##] [weight] [italic][strikeout]}}`

Example:

```
 {{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.

22.40 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 OpenText Fax\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.

You can also use this code to specify that no form should be overlaid.

Syntax

```
 {{formname coord}}
```

Example:

```
 {{form order 0.5 0.2}}
 {{form none}}
```

The first example sets the form to

OpenText Fax\Production\Forms\order

or

OpenText Fax\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 238](#)
[“UNITS” on page 256](#)

22.41 FROM

The sender's email address. This command is only available if you have licensed the InternedLink Module.

Syntax

`{{fromemailaddress}}`

Example:

```
{{from accounting@opentext.com}}
```

See also

["TO" on page 252](#)

["CC" on page 212](#)

["SUBJECT" on page 250](#)

22.42 IMAGETYPE

Selects the graphic format of a {{type mime}} document. The default graphic format for {{type mime}} documents is Group 4 TIFF. The searchable PDF type (SPDF) requires the OpenText OpenText Fax Searchable PDF Module. If the module isn't licensed, then non-searchable PDFs will be created.

Syntax

`{{imagetype{pdf|spdf|landscape|group3|group4|pcx}}}`

Example:

```
{{imagetype pdf landscape}}
```

The example identifies PDF in landscape orientation as the format for a {{type mime}} document.

See also

["TYPE" on page 252](#)

22.43 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 default is OpenText Fax\Production\Include. If the include file cannot be found, the OpenText Fax Integration Module tries again, adding .inc to the file name.

Syntax

`{{includefilename}}`

Example:

```
 {{include setup}}
```

The example includes input from the file OpenText Fax\Production\Include\setup. If that file does not exist, then it would include input from the file OpenText Fax\Production\Include\setup.inc.



Note: Newly-created include files may not be immediately available due to Windows delays. If this occurs, try placing the {{include}} commands in a separate document, and submit it after the document that uses the {{file}} or {{list}} command to create the file.

22.44 LIBDOC

Attaches the specified OpenText Fax library document.

Syntax

```
 {{libdocID}}
```

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. OpenText Fax stores these in the OpenText Fax\Image folder. For more information on library documents, see *OpenText Fax - Administrator Guide (FXNET240400-AGD)*.

22.45 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]}}
```

Example:

```
 {{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 227](#)

["LINEWIDTH" on page 227](#)
["RLINE" on page 245](#)
["RLINETO" on page 245](#)
["UNITS" on page 256](#)

22.46 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

["LINEWIDTH" on page 227](#)
["RLINE" on page 245](#)
["RLINETO" on page 245](#)
["UNITS" on page 256](#)

22.47 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
`{{linewidthpoints}}`

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 226](#)

22.48 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 OpenText Fax\Production\Include. This process will replace a file of the same name. {{List}} differs from {{file}} only in that it strips any leading white space or blank lines from the input. {{List}} is useful in creating and replacing broadcast lists.

Syntax

`{{listfilename}}`

Example:

```
{list bcast.inc}
```

The example writes subsequent input (until an {{end}} command) to OpenText Fax\Production\Include\bcast.inc.



Note: Newly-created include files may not be immediately available due to Windows delays. If this occurs, try placing the {{list}} command in a separate document, and submit it before the document that uses the {{include}} command to reference the file.

See also

[“FILE” on page 221](#)

22.49 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 211](#)

[“TM” on page 251](#)

[“UNITS” on page 256](#)

22.50 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, email 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 inserts “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 inserts the default information. If you do not specify the path to the lookup table, the default is `OpenText Fax\Production\Include`.

22.51 LP, LPR, or PRINTER

Sets the default printer for the print FCL commands such as `{type print}` and `{printnow}`.

Syntax

`{lp|lpr|printername}`

Example:

```
{lp local}
```

The example sets the printer name to local.

See also

[“PRINTER” on page 241](#)
[“TYPE” on page 252](#)

22.52 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`

Example:

```
{ {{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 240](#)
[“RMOVETO” on page 246](#)
[“UNITS” on page 256](#)

22.52.1 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 to support the command.

On the OpenText Fax server, edit the Windows registry

1. Go to the subkey HKEY_LOCAL_MACHINE\Software\RightFax\Parse.
2. Add the REG_MULTI_SIZE registry value UnixSpecials.
3. Enter the string MoveTo in the multi-string editor.

22.53 NOCOVER or NO_COVER

Same as {{cover none}} ([“COVER” on page 215](#)). No cover sheet is generated, even if you have defined a default cover sheet in the OpenText Fax Integration Module Configuration program.

Syntax

`{{nocover}}`

Example:

```
{{nocover}}
```

The example allows the document to be processed with no cover sheet.

22.54 NOTE

Same as {{“REM” on page 243}}. Inserts any information; is commonly used for troubleshooting. It has no effect on the document.

Syntax

`{{notetext}}`

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.

22.55 NOTIFYHOST

Specifies the templates to be used to format the success or failure of the transmission of the document. It also can specify which channel ID to use to send the notifications. The template files (in the syntax below, these are `Successtemplate.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, number 16, will be used.

Syntax

`{{notifyhost successtemplate failuretemplate [channel]}}`

Example:

```
{{notifyhost mysucc myfail mynotify}}
```

The example shows that, if the document is sent successfully, the OpenText Fax Integration Module generates a notification message using the `OpenText Fax\Production\Include\Mysucc`

or

OpenText Fax\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 channel.

See also

[“ONERROR” on page 232](#)
[“ONSUCESS” on page 234](#)

22.56 ONERROR

Describes what the OpenText Fax Integration Module does in the event the document fails during transmission. This command overrides the settings you establish in the OpenText Fax Server Configuration program for each document in which you use the {{onerror}} command. See [“Creating notification templates” on page 95](#).



Note: This command specifies actions based on transmission failures only. It does not apply to transmissions that are blocked by a dialing rule.

Options for {{onerror}} are fax, delete, email (or mime), certified, nothing, or print.

Option	Description
Certified	If a transmission error occurs in faxing, the document will be sent as a certified email document. This option requires that you have licensed the SecureDocs Module. You can configure a notification to alert you that a document failed as a fax and was sent as a certified email. See “Creating notification messages with FCL” on page 93 .
Delete	If a transmission error occurs in faxing, the fax image will be deleted. If you use {{onerror delete}}, then successful faxes will also be deleted.
Email or mime	If a transmission error occurs in faxing, the document will be sent via email. See “Creating FCL documents with InternetLink commands” on page 187 .

Option	Description
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 OpenText Fax Configuration program. See "Setting defaults for FCL documents" on page 162.</p>
Nothing	<p>If a transmission error occurs in faxing, no special action is taken. This overrides any defaults you set with the OpenText Fax Configuration program. See "Setting defaults for FCL documents" on page 162.</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 OpenText Fax Configuration program. See "Setting defaults for FCL documents" on page 162.</p>

Syntax

```

{{onerror nothing}}
{{onerror delete}}
{{onerror fax number delete}}
{{onerror email | mime | certifiedaddress delete}}

```

```

{{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 emailed to css@oswego.com via the InternetLink Module.

The third example shows that, upon failed transmission, the document will be emailed to css@oswego.com as a certified document via the SecureDocs Module.

For more information on the SecureDocs Module, see *OpenText Fax - SecureDocs Module Administrator Guide (FXNET240400-ASD)*.

See also

["NOTIFYHOST" on page 231](#)

[“ONSUCCESS” on page 234](#)

`{{Onerror delete}}` translates to the OpenText Fax server field `faxflag_autodeleteall`. See [“Creating and attaching cover sheets and other files” on page 61](#)

22.57 ONSUCCESS

Describes what the OpenText Fax Integration Module does in the event the document transmits successfully. This command overrides the settings you establish in the OpenText Fax Server Configuration program for each document in which you use the `{{onsuccess}}` command. See [“Setting defaults for FCL documents” on page 162](#).

Options for `{onsuccess}` are fax, delete, email (or mime), certified, nothing, or print.

Option	Description
Certified	When a transmission succeeds in faxing, the document will also be sent as a certified email document. This option requires that you have licensed the SecureDocs Module. You can configure a notification to alert you that a document was sent as a certified email. See “Creating notification messages with FCL” on page 93
Delete	When a transmission succeeds in faxing, the fax image will be deleted.
Email or mime	When a transmission succeeds in faxing, the document will also be sent via email. You can configure a notification to alert you that a document was sent as an email. See “Creating notification messages with FCL” on page 93
Fax	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). If you don’t enter a fax number, the software uses the default number specified in the OpenText Fax Configuration program. See “Setting defaults for FCL documents” on page 162 .

Option	Description
Nothing	When a transmission succeeds in faxing, no special action is taken. This overrides any defaults you set with the OpenText Fax Configuration program. See “ Setting defaults for FCL documents ” on page 162.
Print	When a transmission succeeds in faxing, the document will print. Enter a printer ID defined in Enterprise Fax Manager. If you don’t enter a printer, the software uses the default printer specified in the OpenText Fax Configuration program. See “ Setting defaults for FCL documents ” on page 162.

Syntax

```

{{onsuccess nothing}}
{{onsuccess delete}}
{{onsuccessfax number delete}}
{{onsuccess email | mime | certifiedaddress delete}}

```

```

{{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 emailed to css@oswego.com via the InternetLink Module.

The third example shows that, after successful fax transmission, the document will be emailed to css@oswego.com as a certified document via the SecureDocs Module.

For more information on the SecureDocs Module, see *OpenText Fax - SecureDocs Module Administrator Guide (FXNET240400-ASD)*

See also

[“NOTIFYHOST” on page 231](#)
[“ONERROR” on page 232](#)

`{{Onsuccess delete}}` translates to the OpenText Fax server field `faxflag_autodelete`. See [“Using cover sheets in a broadcast” on page 167](#).

22.58 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.

22.59 OWNER

Specifies the document owner's name.

Syntax

```
 {{ownername}}
```

Example:

```
 {{owner Jane Doe}}
```

The example specifies the owner of this document as "Jane Doe."

{{Owner}} translates to the OpenText Fax server field from_name. See "[Using cover sheets in a broadcast](#)" on page 167.

22.60 PDFTYPE

Specifies whether attached PDF files consist primarily of text, photos or images, or are light in color. The quality of the attached PDF files is adjusted according to the setting.

Syntax

```
 {{pdftype text}}
```

```
 {{pdftype photoimage}}
```

```
 {{pdftype light}}
```

Example

```
 {{begin}}
 {{fax 1234}}
 {{Contact John}}
 {{pdftype light}}
 {{attach C:\User\rhondasmith\Desktop\reports\report1.pdf}}
```

```
Call Dr. Miller.  
{{end}}
```

The above example specifies that the report1.pdf file is light in color and should be darkened.

22.61 PDFOWNER

For use when sending a protected PDF through the SecureDocs module, this command specifies a password that the recipient must enter to open the PDF in Adobe Reader.

Syntax

```
{{pdfowner "password"}}  
{{pdfowner"Op3nSesame"}}
```

22.62 PDFUSER

For use when sending a protected PDF through the SecureDocs module, this command specifies the password and permission levels for the PDF.

Syntax

```
{{pdfuser name p}}
```

Example

```
{{pdfuser "#MNid4" 31}}
```

In the example, "#MNid4" is the user's password, and the user has full permissions.

Permission is an integer bit-wise value that grants certain permissions for the user as described in the following table.

Bit	Permission	Notes
1	Print	The user can print the PDF. Example: {{pdfuser "#MNid4" 1}}
2	Edit	The user can edit the PDF. Example: {{pdfuser "#MNid4" 3}} This example sets print and edit permissions.
4	Select	The user can select and copy text. Example: {{pdfuser "#MNid4" 7}} This example sets print, edit, and select permissions.

Bit	Permission	Notes
8	Annotate	The user can add annotations. Example: {{pdfuser "#MNid4" 15}} This example sets print, edit, select, and annotate permissions.
16	Save As	The user can save the PDF to a new file name. Example: {{pdfuser "#MNid4" 31}} This example sets print, edit, select, annotate, and save as permissions.

22.63 PLACE

Positions the specified Class F TIF image on the page. The file can be specified using the full path or relative to OpenText Fax\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

{{placegraphiccoord}}

Example:

```
  {{place yoyodyne.tif 5 4}}
```

The example places the graphic image Yoyodyne.tif on the page at (5,4) in specified units that you set in the OpenText Fax Integration Module Configuration program. See “Setting defaults for FCL documents” on page 162.

See also

[“PLACEXY” on page 240](#)
[“BEGINCVT” on page 206](#)

22.64 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 OpenText Fax\Production\Forms. You can also specify x- and y-coordinates 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}}
```

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 OpenText Fax Integration Module Configuration program. See “[Setting defaults for FCL documents](#)” on page 162.

See also

[“PLACE” on page 238](#)
[“PLACEXY” on page 240](#)

22.65 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 OpenText Fax\Production\Forms. You can also specify x- and y-coordinates 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. {{Placelast}} 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 238](#)
[“PLACEXY” on page 240](#)

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 OpenText Fax Integration Module Configuration program. See [“Setting defaults for FCL documents” on page 162](#).

22.66 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 OpenText Fax 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 238](#)

22.67 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 230](#)
[“RMOVETO” on page 246](#)

22.68 PREVIEW

Holds the document for preview in the FaxUtil mailbox.

Syntax

```
{{preview}}
```

Example:

```
{{preview}}
```

22.69 PRINTER

See

[“LP, LPR, or PRINTER” on page 229](#)

22.70 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.

Example:

```
{{printnow print 2}}
```

The example immediately prints two copies of the document.

See also

[“TYPE” on page 252](#)

[“LP, LPR, or PRINTER” on page 229](#)

22.71 PRIORITY

Specifies the priority at which the document will be processed and scheduled. High priority documents are 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 217](#)
[“TIME” on page 251](#)

`{{Priority}}` translates to the OpenText Fax server field **ucPriority**.

22.72 PRIVATEFAX

Specifies the sender's private fax number.

Syntax

`{{privatefax5551212}}`

Example:

```
 {{privatefax 551212}}
```

The example places the fax number 555-1212 on the fax cover sheet.

22.73 RBOX

Draws a box in the current line width at the specified coordinates. It can also fill 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"}}`

Example:

```
 {{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 212

22.74 REM

Same as "NOTE" on page 231. Inserts any information; is commonly used for troubleshooting. It has no effect on the document.

Syntax

`{{remtext`

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.

22.75 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 OpenText Fax XML Interface. REPLYTO is the field in the submit post that the XML Interface populates to determine where to send the notification.

Syntax

`{{replytorecipient`

Example:

```
 {{replyto www.opentext.com}}
```

The example specifies that the XML notification should return to www.opentext.com.

22.76 RETRYCOUNT

Controls the number of times the server will retry sending a fax.

Valid entries are 1 to 255. You must also set a value for the RETRYINTERVAL command if you set a value for RETRYCOUNT.

You can place this command anywhere in the FCL document.

Syntax

`{{retrycount <n>}}`

Example:

```
 {{retrycount 4}}
```

This example sets the retry count to 4.

22.77 RETRYINTERVAL

Controls the interval (in minutes) that the server will wait between each attempt to resend a fax.

Valid entries are 1 to 255. You must also set a value for the RETRYCOUNT command if you set a value for RETRYINTERVAL.

You can place this command anywhere in the FCL document.

Syntax

```
{{retryinterval <n>}}
```

Example:

```
{{retryinterval 7}}
```

This example sets the interval between attempts to 7 minutes.

22.78 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 212](#)

[“FILLBOX” on page 222](#)

22.79 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>)]}}
```

Example:

```
 {{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 226](#)
[“UNITS” on page 256](#)

22.80 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 226](#)
[“LINETO” on page 227](#)
[“LINEWIDTH” on page 227](#)
[“RLINE” on page 245](#)
[“UNITS” on page 256](#)

22.81 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

`{}{rmovetocoord}`

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 230](#)
[“POSITION” on page 240](#)
[“UNITS” on page 256](#)

22.82 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 [“STARTPOLY” on page 249](#). 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

`{}{rstartpolycoord}`

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. Establish units, such as inches or centimeters, in the OpenText Fax Integration Module Configuration program. See [“Setting defaults for FCL documents” on page 162](#).

See also

[“ENDPOLY” on page 219](#)

[“LINETO” on page 227](#)
[“RLINETO” on page 245](#)
[“STARTPOLY” on page 249](#)
[“UNITS” on page 256](#)

22.83 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.

22.84 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 tabcoord{1|r|c|d}}`

Example:

```
{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

[“CLEAR TABS” on page 213](#)
[“TAB” on page 250](#)

22.85 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 define {{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 248](#)

22.86 SIGNATURE

Specifies the name of a graphic file of a signature that appears 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 248](#)

22.87 SMS

Specifies the phone number of the SMS-capable device that will receive notifications about the fax transmission

Syntax

{{{smsspageID}}}

Example:

```
 {{sms 520-555-1212}}
```

See also

[“TYPE” on page 252 \(SMS\) and “SMSMSG” on page 249](#)

22.88 SMSMSG

Specifies the text of the SMS notification to send

Syntax

`{smsmsgtext}`

Example:

```
{smsmsg Fax was sent successfully}
```

See also

[“TYPE” on page 252 \(SMS\)](#)

[“SMS” on page 248](#)

22.89 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}` ([“RSTARTPOLY” on page 246](#)).

- `{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 OpenText Fax Server Configuration program. See [“Setting defaults for FCL documents” on page 162](#).
- `{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

`{startpolycoord}`

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 219](#)
[“RSTARTPOLY” on page 246](#)
[“UNITS” on page 256](#)

22.90 SUBJECT

The subject line for an email. This command is only available if you have licensed the InternedLink Module.

Syntax

```
 {{subject topic}}
```

Example:

```
 {{subject January invoices}}
```

See also

[“TO” on page 252](#)
[“CC” on page 212](#)
[“FROM” on page 225](#)

22.91 TAB

Creates a single tab stop based on the information you create with the {{settab}} command.

Syntax

```
 {{tab tab stop text|number}}
```

Example:

```
 {{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 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” appear aligned left at 1.5 inches from the left margin.

See also

[“CLEAR TABS” on page 213](#)
[“SETTAB” on page 247](#)

22.92 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.

22.93 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 is 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 OpenText Fax 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 217](#)

22.94 TM

Sets the top margin for the current and subsequent pages in the specified units. After this command, text above this margin will not render. 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 211](#)
[“LM” on page 228](#)

22.95 TO

The recipient’s email address. This command is only available if you have licensed the InternedLink Module.

Syntax

`{{to emailaddress}}`

Example:

```
 {{to accounting@opentext.com}}
```

See also

[“FROM” on page 225](#)
[“CC” on page 212](#)
[“SUBJECT” on page 250](#)

22.96 TRANID

Sets the identification of the transaction that produced the document.

Syntax

`{{tranid ID}}`

Example:

```
 {{tranid BR549}}
```

The example sets the transaction ID for this document to “BR549”.

22.97 TYPE

This specifies the type of document. You must specify the document type with either this command, the {{fax}} command for faxes, or the {{begin}} command for email and SMS. The available document types are listed here.

Option	Description
Fax	Documents are rendered as TIF images and transmitted via fax.

Option	Description
Print	<p>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 includes production cover sheets (.cov files). Enterprise cover sheets (.pcl and .doc files) are not printed.</p>
File	<p>Documents are rendered as TIF images. You can specify the file and path names. This command cannot be used when submitting Embedded or False First Page (FFP) documents to the Integration Module.</p>
Email	<p>Documents are rendered as text and included in the body of an email message. Note that if you have specified email format in the {{begin}} command, you do not need to use the type command.</p>
Mime	<p>Documents are rendered as TIF images and included as an attachment to an email 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, PDF, or SPDF).</p> <p>Note that if you have specified mime format in the {{begin}} command, you do not need to use the type command.</p>
Certified	<p>Documents are sent as certified emails. To send documents certified delivery, you must license and install the OpenText Fax SecureDocs Module.</p>

Option	Description
SMS	<p>Notifications about the fax transmission are sent to a OpenText Fax user's SMS-capable device.</p> <p>Use the syntax {{type sms<serviceid>}} where <serviceID> is the service ID for an SMS service defined in OpenText Fax.</p> <p>To specify the SMS telephone number that will receive the text, use the {{sms}} FCL code. Use the syntax {{type sms <nnnnnnnnnn>}}, where <nnnnnnnnnn> is the recipient's SMS telephone number. You can enter more than one number separated by semicolons.</p> <p>For more information on defining SMS services for internal notification messages, see <i>OpenText Fax - Administrator Guide (FXNET240400-AGD)</i>.</p>
SMSDoc	<p>The text in the FCL command line is sent externally as an SMS text message via DocTransport.</p> <p>Use the syntax {{type smsdoc}}.</p> <p>To specify the SMS telephone number that will receive the text, use the {{sms}} FCL code. Use the syntax {{type sms <nnnnnnnnnn>}}, where <nnnnnnnnnn> is the recipient's SMS telephone number. You can enter more than one number separated by semicolons.</p> <p>To specify the text of the SMS message, use the {{smsmsg}} FCL code. Use the syntax {{smsmsg <text>}}, where <text> is the text of the message.</p> <p>For more information on defining SMS services in a DocTransport, see <i>OpenText Fax - Administrator Guide (FXNET240400-AGD)</i>.</p>

Syntax

{{type type copies filename}}

Example:

```
 {{type fax}}
 {{type file c:\Program Files\OpenText Fax\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 OpenText Fax 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. See “[Setting defaults for FCL documents](#)” on page 162.

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 OpenText Fax Server Configuration program (it must also be defined in Enterprise Fax Manager). To change the printer with FCL, you can include {{lp}}, {{pr}}, or {{printer}} (“[LP, LPR, or PRINTER](#)” on page 229).

See “[Setting defaults for FCL documents](#)” on page 162.

See also

[“PRINTNOW” on page 241](#)
[“LP, LPR, or PRINTER” on page 229](#)

22.98 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 222](#)

22.99 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 *OpenText Fax - Administrator Guide (FXNET240400-AGD)*.) 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}}`

Example:

```
 {{unique_id test:01ea}}  
 {{unique_id test:01eb}}
```

`{{Unique_id}}` translates to the OpenText Fax server field `unique_id`.

22.100 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 OpenText Fax Server Configuration program. See [“Setting defaults for FCL documents” on page 162](#). 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.

22.101 USER1, USER2, or USER3

These commands can hold user-defined information.

These commands can be used only in notifications and OpenText Fax Integration Module (.cov) cover sheets.

Syntax

```
 {{user1 user information}}  
 {{user2 user information}}  
 {{user3 user information}}
```

Example:

```
 {{user1 Some important information}}
```

22.102 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”.

22.103 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.

Universal Coordinated Time is assumed unless you specify 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.

Syntax

```
{{UTC YYYY{- | /}MM{- | /}DD{T | }hh:mm:ss TZD}}
```

Example:

```
{{UTC 2000-01-24T22:25:00Z}}
{{UTC 2000-01-24T23:25Z+01:00}}
{{UTC 2000-01-24T14:25-08:00}}
```

Represent times as two digits, with leading zeros as necessary, with the exception of the four-digit year and the TZD.

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)	

22.104 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 OpenText Fax server field `to_contactnum`.

See Also

[“Using cover sheets in a broadcast” on page 167](#).

22.105 WINSECID

Specifies the OpenText Fax user. By default, `{{winsecid}}` is configured to automatically create users when faxes are sent from non-existent users. You can disable this setting so that users are not created. If the user ID specified in `{{winsecid}}` does not exist, the fax fails and an error is logged.



Note: Because `{{winsecid}}` can create a user, it is different from “[USERID](#)” [on page 257](#), which is informational.

Place the `{{winsecid}}` command in the `global.beg` file or in the FCL from the host. Without a `{{winsecid}}` command, the default user account will be used to send documents from the OpenText Fax Server. The settings for a default user are typically not optimized for sending documents from the OpenText Fax Server (notifications, cover sheet settings, etc.).

By changing the `{{winsecid}}` command in the `global.beg` file, you can specify or change the default OpenText Fax user account used by the OpenText Fax Server. (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}}
```

22.105.1 Controlling the creation of users for faxes sent from non-existent users

To enable or disable the automatic creation of users

1. On the Start menu, click **Programs**, **OpenText**, and then **OpenText Fax Enterprise Fax Manager**.
2. In the **Fax Servers** list, click the name of the server on which the OpenText Fax Integration Module is running.
3. In the **Service Name** list, double-click **OpenText Fax Integration Module**.
4. In the left pane of the **Integration Module Configuration** window, under **General**, click **More Options**.
5. Do one of the following:
 - To disable the automatic creation of users, clear the **Allow automatic creation users** check box. When the user ID specified in {{winsecid}} does not exist, the fax fails and an error is logged.
 - To enable the automatic creation of users, select the **Allow automatic creation users** check box. When the user ID specified in {{winsecid}} does not exist, a new user is created and the fax is sent.

