Captaris Proprietary

This material is Captaris Class A Proprietary and is provided to assist our customers in understanding and utilizing Captaris products.

Copyright © 2002 Captaris, Inc.
All rights reserved.
Captaris is a registered trademark of Captaris, Inc.

Release 2.1
February 4, 2002
Printed in U S America
030-80052-00
## Revision Control Page

<table>
<thead>
<tr>
<th>Revision Level</th>
<th>Date Issued</th>
<th>Revision Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>R2.0</td>
<td>1/24/02</td>
<td>Initial Release; updated to reflect the RightFax 8.0 enhancements.</td>
</tr>
<tr>
<td>R2.1</td>
<td>2/4/02</td>
<td>Adjustments made to Sections 1, 3, 5, 6, and 8 following the first teach.</td>
</tr>
</tbody>
</table>
Contents

Course Description ................................................................. ix

About This Course ............................................................... xi

Purpose .............................................................................. xi
How to Use This Document ................................................ xi
Organization ..................................................................... xi

Section 1. XML - Fax

Objectives .......................................................................... 1–1
XML-Fax Overview ............................................................ 1–2
XML Document Flow ......................................................... 1–3
XML-Fax Installation ........................................................... 1–5
XML-Fax Transport Methods ............................................. 1–5
ParseXML Command .......................................................... 1–8
  ParseXML options xmlfile attachments ......................... 1–8
Adding Input Devices for Each XML Type ....................... 1–9
XML Schemas ..................................................................... 1–10
  XML Submit Schema ....................................................... 1–11
Sending a Fax Using XML .................................................. 1–16
Common Elements of Submit Schema ............................... 1–18
Using the XML_FAX_SUBMIT schema ............................. 1–19
XML Submit Stylesheet ....................................................... 1–20
  XML Submit Reply DTD .................................................... 1–23
  XML Query Schema ........................................................ 1–26
Querying RightFax Using an XML Document .................... 1–27
Common Elements of Query Schema ............................... 1–29
  Using the XML_FAX_QUERY_schema ............................... 1–30
  XML Query Reply DTD .................................................... 1–31
  XML Action Schema ........................................................ 1–32
Taking Action on a Fax Using XML .................................... 1–33
Common Elements of Action Schema ............................... 1–34
  Using the XML_FAX_ACTION schema ............................... 1–35
  XML Action Reply DTD .................................................... 1–36
Receiving Notification Through XML ............................... 1–37
Section 2. Java Interface to XML Fax

Objectives ................................................................. 2–1
Overview ................................................................. 2–1
Requirements ......................................................... 2–2
Installation ............................................................... 2–2
Submitting a Fax Through Java ................................ 2–3
RFaxSubmit Class .................................................... 2–7
RFaxSubmit and RFDoc Classes ................................. 2–8
Querying the Fax Server .......................................... 2–10
RFaxQuery Class ....................................................... 2–12
Taking Action Against A Fax ................................. 2–14
RFaxAction Class ...................................................... 2–16
Exercise ................................................................. 2–17

Section 3. Understanding the COM Module

Objectives ................................................................. 3–1
Overview ................................................................. 3–2
The RightFax COM API ............................................. 3–2
Understanding Objects, Properties, and Methods ....... 3–2
The FaxServer Object .............................................. 3–3
Enumerated Data Types ........................................... 3–3
The Object Hierarchy ................................................. 3–3
Exploring the COM Object Library Using the Visual Basic Object Browser ...................................... 3–5
Exercise ................................................................. 3–6

Section 4. Creating a Simple Application

Objectives ................................................................. 4–1
Steps to Design a Simple Application ...................... 4–2
Step 1 - Start a New Visual Basic Project ................. 4–2
Step 2 - Set Up the Project Form ......................... 4–3
Step 3 - Create and Log On to a New Fax Server Object ................................................................. 4–4
Step 4 - Display the Users in the List Box .............. 4–6
Step 5 - List Faxes for the Selected User ................. 4–7
Step 6 - Add an Exit Button .................................... 4–9
Step 7 - Test and Compile the Program .................. 4–9
Exercise ................................................................. 4–11
Contents

Section 5. The FaxServer and ServerInfo Objects

Objectives ................................................................. 5–1
FaxServer Object ....................................................... 5–2
FaxServer ................................................................. 5–2
ServerInfo ............................................................... 5–7
Exercise ................................................................. 5–14

Section 6. Fax Objects and Attachments

Objectives ................................................................. 6–1
Fax Object Overview ................................................ 6–2
Attachment .............................................................. 6–2
Attachments Collection ........................................... 6–4
BillingCode ............................................................... 6–5
BillingCodes Collection ........................................... 6–7
CoverSheet .............................................................. 6–10
CoverSheets Collection ............................................. 6–10
Fax ............................................................................. 6–11
Exercise 1 ................................................................. 6–27
Faxes Collection ...................................................... 6–28
Form ......................................................................... 6–29
Forms Collection ..................................................... 6–31
LibraryDocument ..................................................... 6–32
LibraryDocuments Collection ................................. 6–36
Exercise 2 ................................................................. 6–37

Section 7. Users, Groups, and Folders

Objectives ................................................................. 7–1
User Object Overview .............................................. 7–2
Delegate ................................................................. 7–2
Delegates Collection ............................................... 7–5
Folder ..................................................................... 7–6
Folders Collection .................................................. 7–6
Group ..................................................................... 7–7
Groups Collection ................................................... 7–16
Signature ................................................................. 7–17
Signatures Collection .............................................. 7–18
User ......................................................................... 7–19
Users Collection ...................................................... 7–39
Exercise 1 ................................................................. 7–40
## Index

### Section 8. Phonebooks and Printers

<table>
<thead>
<tr>
<th>Phonebooks and Printers</th>
<th>8–1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Phonebook and Printer Objects Overview</td>
<td>8–2</td>
</tr>
<tr>
<td>PhoneBook Collection</td>
<td>8–2</td>
</tr>
<tr>
<td>Phonotel</td>
<td>8–3</td>
</tr>
<tr>
<td>PhonotelElement</td>
<td>8–5</td>
</tr>
<tr>
<td>PhonotelGroup Collection</td>
<td>8–6</td>
</tr>
<tr>
<td>Printer</td>
<td>8–7</td>
</tr>
<tr>
<td>Printers Collection</td>
<td>8–10</td>
</tr>
<tr>
<td>PrintRequest</td>
<td>8–11</td>
</tr>
</tbody>
</table>

### Section 9. Fax History Objects

<table>
<thead>
<tr>
<th>Objectives</th>
<th>9–1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fax History Object Overview</td>
<td>9–2</td>
</tr>
<tr>
<td>FaxHistories Collection</td>
<td>9–2</td>
</tr>
<tr>
<td>FaxHistory</td>
<td>9–2</td>
</tr>
<tr>
<td>ApprovalFaxHistory</td>
<td>9–4</td>
</tr>
<tr>
<td>ConversionErrorFaxHistory</td>
<td>9–4</td>
</tr>
<tr>
<td>DisapprovalFaxHistory</td>
<td>9–5</td>
</tr>
<tr>
<td>FileRouteFaxHistory</td>
<td>9–5</td>
</tr>
<tr>
<td>NetForwardFaxHistory</td>
<td>9–6</td>
</tr>
<tr>
<td>OCRFaxHistory</td>
<td>9–6</td>
</tr>
<tr>
<td>PrintFaxHistory</td>
<td>9–7</td>
</tr>
<tr>
<td>RouteFaxHistory</td>
<td>9–7</td>
</tr>
<tr>
<td>SecureDocHistory</td>
<td>9–8</td>
</tr>
<tr>
<td>TransmissionFaxHistory</td>
<td>9–8</td>
</tr>
<tr>
<td>Exercise 1</td>
<td>9–12</td>
</tr>
</tbody>
</table>

### Section 10. Event Handling and Event Objects

<table>
<thead>
<tr>
<th>Objectives</th>
<th>10–1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Using the Event Handler</td>
<td>10–2</td>
</tr>
<tr>
<td>Creating a Sample Project Using the Event Handler</td>
<td>10–2</td>
</tr>
<tr>
<td>ArchiveEvent</td>
<td>10–3</td>
</tr>
<tr>
<td>CompleteEvent</td>
<td>10–4</td>
</tr>
<tr>
<td>Events</td>
<td>10–5</td>
</tr>
<tr>
<td>MessageEvent</td>
<td>10–6</td>
</tr>
<tr>
<td>ValidateEvent</td>
<td>10–8</td>
</tr>
<tr>
<td>Exercise 1</td>
<td>10–9</td>
</tr>
</tbody>
</table>

### Appendix A. RightFax Enumerations
Course Description

Integrating RightFax

Objectives:

At the completion of this course, the student should be able to:

• Send faxes, query the RightFax server and take action on faxes using XML.
• Retrieve XML reply information.
• Send faxes, query the RightFax server and take action on faxes using Java.
• Retrieve Java reply information.
• Use Visual BASIC to access the COM API and develop fax enabled applications.

Audience:

This course is designed for RightFax integrators and developers.

Prerequisites

Completion of RightFax Technical and Production System Technical training.
Understanding of XML, Java, and Visual BASIC programming environments

Key Topics:

• Creating XML submit, query and action documents.
• Creating Java applications to fax a document, query the server and take action on a fax.
• Creating Faxing applications in Visual BASIC using the COM API.

Integrating RightFax

Course Length: 3 Days
About This Course

Purpose

This is a structured three-day program designed to develop attendee competency in integrating the RightFax Production Suite with third party applications. Attendees will become familiar with RightFax implementation of XML and Java as well as designing applications using the COM API in Visual Basic.

How to Use This Document

This document provides materials to enable the participant to follow the presentations, exercises, and discussions that comprise the course.

Organization

This document is divided into 10 sections and 1 appendix.

Section 1 – XML in the Integration Module
Section 2 – Java
Section 3 – Understanding the COM Module
Section 4 – Creating a Simple Application
Section 5 – The FaxServer and ServerInfo Objects
Section 6 – Fax Objects and Attachments
Section 7 – Users, Groups, and Folders
Section 8 – Phonebooks and Printers
Section 9 – Fax History Objects
Section 10 – Event Handling and Event Objects
Appendix A – RightFax Enumerations
About This Course
Section 1
XML - Fax

Objectives

Upon completion of this section, you should be able to:

- Create an input device for each of the XML document types.
- Submit faxes to the RightFax server using the XML_FAX_SUBMIT schema.
- Query the RightFax server using the XML_FAX_QUERY schema.
- Perform an action on a fax image using the XML_FAX_ACTION schema.
XML-Fax Overview

The XML component of the RightFax Integration Module allows you to submit XML documents for conversion to fcl and delivery to the RightFax server for processing. These documents can be submitted via a number of different methods, however the documents must conform to the schemas provided.
XML Document Flow

1. XML Input
2. Determine XML Type
3a. Submit
4a. Apply XSLT
5a. Submit FCL to RightFAX
6a. Send Fax
7a. Create XML Notification
8a. Send Notification To Sender

1. QUERY
2. ACTION
3b. QUERY RF Database and Return search Results
4b. Update RF Database
4c. Update RF Database
3c. Create XML Notification
8a. Send Notification To Sender
1. The XML document is submitted through the system via HTTP, FTP or IBM MQ Series.

2. Based upon the setting of the ParseXML command in the input device the XML-Fax application determines whether the document is of a submit, query or action type

**If the document is a Submit type:**

3a. An XSLT is applied to the XML document to convert the elements to FCL commands.

4a. The FCL document is submitted to RightFax.

5a. RightFax sends the document

6a. If the original XML document had a notifyhost element a notification is generated

7a. A notification is sent to the sender of the document or however it is specified in the Production environment output device.

**If the document is a Query type:**

3b. The RightFax database is queried and the results are returned as an XML document either through an HTTP reply or as a file specified by the ParseXML command.

**If the document is an Action type:**

3c. The fax specified is either deleted, forwarded or used to create a library document and the database is updated.
XML-Fax Installation

To install XML-Fax:

1. Point to the XML subdirectory on your installation CD or the path specified by your instructor.
2. Double click the setup executable.
3. Click next at the introduction splash screen.
4. Agree to the license agreement by clicking yes.
5. The installation files will be copied to your system and the World Wide Web Publishing service will be stopped if it is running (after the files have been installed the service will be restarted).
6. When the RightFax XML-Fax Setup Dialog box appears click finish.

XML-Fax Transport Methods

The Integration Module XML interface supports three methods of transport. The methods available for transport are:

1. HTTP / HTTPS
2. File
3. IBM MQSeries

We will use the file method of transport to move their XML documents from their legacy systems into the RightFax Integration Module. For information on the HTTP / HTTPS and IBM MQSeries transport methods please reference the RightFax 8.0 Integration Module guide.

By selecting file as the transport method we are able to create an XML document and ftp it to a directory that the Integration Module will monitor. When the document is delivered the XML-Fax Module will parse the file using the parseXML executable and output FCL to the Integration Module for processing. You will need to add a BufDir input device in the Integration Module control panel applet to monitor the directory that we will place the XML documents into.
To create this input device:

1. Open Enterprise Fax Manager and double click on the Integration Module service.

2. Select Inputs and click Add Input Device.
3. Select Directory as the input type.
4. Name the input device XMLSubmit
5. In the Folder to Monitor text box type
   D:\xmlsubmit
6. In the Execute on Input text box type:
   `parsexml -f "d:\xmlsubmit\reply\out.xml"` $$
7. Click Ok.
ParseXML Command

The parseXML command parses the XML document your system provides and converts it to FCL for delivery to the RightFax Integration Module. This command is an executable application that can be run from a DOS prompt or combined with an Integration Module input device. The syntax of the command is:

```
ParseXML options xmlfile attachments
```

The available command line options are:

- `q` XML input is query document.
- `a` XML input is action document
- `f <filename>` Output reply XML to the file specified using this option.
- `-?` Command line parameter help.
- `-h` Command line parameter help.

Let's take a look at what has been created in the command line text box of the input device from the previous section. The command was:

```
bufdir –c “parsexml -f “d:\xmlsubmit\reply\out.xml” $$” d:\xmlsubmit
```

1. The bufdir command specifies that the Integration Module will listen to the directory specified (in this case d:\xmlsubmit) for any changes.
2. The –c option specifies that the text that follows (within the quotes) is a command that should be executed.
3. The parsexml command will parse the XML document provided (as specified by the $$ variable).
4. The –f option specifies the file that will be written for the xml reply. In our example the file is out.xml and will be written to the d:\xmlsubmit\reply directory.
5. We will only place XML submit documents in the directory specified because we did not include the a or q option on the command line.
Adding Input Devices for Each XML Type

We will be utilizing all of the XML document types (submit, action, and query). As a result you will need to create a separate input device for each type. Complete the following tasks:

1. Create a bufdir input device for each of the document types. Be sure that you are monitoring a separate directory for each type.

2. What is the syntax of the command you will use for the query document type?

3. Why do you need a separate input device for each of the document types?
XML Schemas

The XML documents submitted by your information system must conform to the XML schemas provided by the Integration Module. There are three functions you can perform as specified in your document. The functions are:

1. **Submit**: This function allows you to submit information to send a fax.
   
   *Note: If you have not installed the XML-Fax application to the default directory (i.e. c:\program files\RightFax\Production\xml) you must change the name space in the stylesheet to reflect the proper location of the schema files as reflected in your XML document (the name space is case sensitive).*

2. **Query**: This function allows you to query RightFax for information related to a fax.

3. **Action**: This function allows you to take action on a specified fax (i.e. forward).

Each of these functions has a reply XML document associated with them that provides return information. The reply document are returned to the system either through an HTTP reply or via the file specified through the –f option of ParseXML.

Each function is associated to a RightFax supplied schema. Each reply conforms to a RightFax supplied DTD. The schemas and DTDs follow:
XML Submit Schema

<?xml version="1.0" encoding="UTF-8"?>
<!-rfxml version="2.0" ->
<Schema name="XML_FAX_SUBMIT" xmlns="urn:schemas-microsoft-com:xml-data"
xmlns:dt="urn:schemas-microsoft-com:datatypes">
  <ElementType name="SEND_DATE_TIME" model="closed" content="textOnly"
dt:type="dateTime.tz"/>
  <ElementType name="INCLUDE_BEG" model="closed" content="textOnly"
dt:type="string"/>
  <ElementType name="ADD_LIBDOC" model="closed" content="textOnly"
dt:type="string"/>
  <ElementType name="INCLUDE_END" model="closed" content="textOnly"
dt:type="string"/>
  <!--SENDER element-->
  <ElementType name="FROM_NAME" model="closed" content="textOnly"
dt:type="string"/>
  <ElementType name="EMP_ID" model="closed" content="textOnly"
dt:type="string"/>
  <ElementType name="FROM_COMPANY" model="closed" content="textOnly"
dt:type="string"/>
  <ElementType name="FROM_DEPARTMENT" model="closed" content="textOnly"
dt:type="string"/>
  <ElementType name="FROM_PHONE" model="closed" content="textOnly"
dt:type="string"/>
  <ElementType name="RETURN_EMAIL" model="closed" content="textOnly"
dt:type="string"/>
  <ElementType name="BILLINFO1" model="closed" content="textOnly"
dt:type="string"/>
  <ElementType name="BILLINFO2" model="closed" content="textOnly"
dt:type="string"/>
  <ElementType name="REPLY_TO" model="closed" content="textOnly" dt:type="uri"/>
  <ElementType name="RF_USER" model="closed" content="textOnly"
dt:type="string"/>
  <ElementType name="SENDER" model="closed" content="eltOnly" order="seq">
    <element type="FROM_NAME" minOccurs="0" maxOccurs="1"/>
    <element type="EMP_ID" minOccurs="0" maxOccurs="1"/>
    <element type="FROM_COMPANY" minOccurs="0" maxOccurs="1"/>
    <element type="FROM_DEPARTMENT" minOccurs="0" maxOccurs="1"/>
    <element type="FROM_PHONE" minOccurs="0" maxOccurs="1"/>
    <element type="RETURN_EMAIL" minOccurs="0" maxOccurs="1"/>
    <element type="BILLINFO1" minOccurs="0" maxOccurs="1"/>
    <element type="BILLINFO2" minOccurs="0" maxOccurs="1"/>
    <element type="REPLY_TO" minOccurs="0" maxOccurs="1"/>
    <element type="RF_USER" minOccurs="1" maxOccurs="1"/>
  </ElementType>
</Schema>
<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"/>
</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"/>
  <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="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"/>
</ElementType>
<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_leading" dt:type="il"/>
    <AttributeType name="font_pitch" dt:type="il"/>
</ElementType>

<ElementType name="FONT" content="empty">
    <AttributeType name="name" dt:type="string" required="yes"/>
    <AttributeType name="leading" dt:type="il"/>
    <AttributeType name="pitch" dt:type="il"/>
</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"/>
</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"/>
</ElementType>
<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>
Sending a Fax Using XML

To send a fax using XML you must create a document which conforms to the XML_FAX_SUBMIT_schema specified.

As the lead engineer at CAF you would like to test the configuration of the XML-Fax system by creating a simple XML document. The XML document you would create might look like this:

```xml
<?xml version="1.0" ?>
<XML_FAX_SUBMIT
 xmlns="d:\RightFax\production\xml\schemas\XMLFAX_SUBMIT_schema.xml"
 stylesheet="XML_FAX_SUBMIT.XSL">
 <DESTINATIONS>
   <FAX unique_id="PRODXML:0001">
     <TO_FAXNUM>1111</TO_FAXNUM>
   </FAX>
 </DESTINATIONS>
 <BODY>
   This is a test of XML Submit
 </BODY>
</XML_FAX_SUBMIT>
```
To test this document:

1. Create a user in the RightFax database named test and give it a routing code of 1111.
2. Open XML Spy on your computer.
4. Specify that you will associate the document with a schema.
5. Browse to the following directory:
   D:\RightFax\production\xml\schemas\XML_FAX_SUBMIT_schema.xml

6. Type in the XML document above as demonstrated by your instructor.
7. Save the XML document to D:\XMLSampleCode\SubmitSample
8. Name the document BasicSubmit.XML
9. Copy BasicSubmit.xml to the directory you specified for submit documents in the previous section.
10. Where can you see if the document was successfully submitted to the system?
Common Elements of Submit Schema

When creating a submit XML document there are a number elements and attributes that are required or used on a regular basis. For your convenience we have listed these elements and attributes with a brief description in parent / child order:

**XML_FAX_SUBMIT** (root element)
- **SEND_DATE_TIME** (Element that specifies date and time to send fax)
- **SENDER** (Parent element to all sender elements)
  - **FROM_NAME** (Element that specifies the name of the sender)
  - **FROM_COMPANY** (Element that specifies the name of the company)
  - **RF_USER** (Element that specifies the senders RightFax user id)
- **DESTINATIONS** (Required parent element of faxes to be sent)
  - **FAX** (Required parent element to recipient elements)
    - **TO_FAXNUM** (Required element that specifies fax number)
    - **TO_NAME** (Element that specifies recipients name)
    - **TO_CONTACTNUM** (Element that specifies recipient phone number)
    - **COVERSHEET** (Element that specifies the coversheet to use)
  - **EMAIL** (Required parent element to email recipient elements)
    - **TO_EMAIL** (Element that specifies the recipient's email address)
  - **PRINT** (Required parent element to print outs)
    - **COPIES** (Element that specifies number of copies to print)
    - **PRINTER_NAME** (Element that specifies name of printer in RightFax)
- **FORM** (Element that specifies the form to be placed on the fax)
- **BODY** (Element that specifies the text to be placed in the body of the fax)
- **ATTACHMENT** (Parent element that specifies a TIF image(s) to be appended to the fax)
  - **FILE** (Empty Element)
  - **path** (File attribute that specifies the location of the TIF image)
Using the XML_FAX_SUBMIT schema

After your test has faxed successfully, the CIO requests that you attempt another test that adds more complexity to your document. To complete this, perform the following exercises:

- Add the FCS.PCL coversheet to a fax, fill in the “to” and “from” info and send it to yourself.
- Create an include file and add formatting instructions (i.e. font command) to it. Utilize the include file in your XML submit document.
- Send a new fax to yourself with the FCS.PCL coversheet and an attachment.
- Add the PO form to your document.
- Add Terms.tif to your document as an attachment.
- Send the document to yourself as an email. Your email address is studentXX@captraining.com. Open outlook and logon using the Outlook profile. Check to see if the email was delivered.

For the next set of exercises complete the following steps:

1. Create three additional users in EFM.
2. Assign each user a unique routing code.
   - Send the completed XML submit document to the three users you created.
XML Submit Stylesheet

The stylesheet which converts the XML Submit document to an FCL document is listed below:

```xml
<?xml version="1.0" encoding="UTF-8"?>
<!--rfxml version="2.0" -->
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
xmlns:xdr="x-
schema:c:\progra-1\rightfax\production\xml\schemas\XML_FAX_SUBMIT_schema.xml">
<xsl:output method="text" encoding="UTF-8" omit-xml-declaration="yes" indent="no" cdata-
section-elements="BODY"/>
<xsl:template match="/">
<xsl:apply-templates select="xdr:XML_FAX_SUBMIT"/>
</xsl:template>
<xsl:template match="xdr:XML_FAX_SUBMIT">
<xsl:text>{{begin}}</xsl:text>
<xsl:if test="@java = 1">{NOTE Document From JAVA API}}</xsl:if>
<xsl:if test="xdr:SEND_DATE_TIME">{utc <xsl:value-of
select="xdr:SEND_DATE_TIME"/>}</xsl:if>
<xsl:if test="xdr:INCLUDE_BEG">{include <xsl:value-of
select="xdr:INCLUDE_BEG"/>}</xsl:if>
<xsl:if test="xdr:FORM">{form <xsl:value-of select="xdr:FORM"/>}</xsl:if>
<xsl:if test="xdr:INCLUDE_END">{include <xsl:value-of
select="xdr:INCLUDE_END"/>}</xsl:if>
<xsl:text>{{end}}</xsl:text>
</xsl:template>
</xsl:stylesheet>
```

- **SENDER**

```xml
<xsl:template match="xdr:SENDER">
<xsl:if test="xdr:FROM_NAME">{owner <xsl:value-of select="xdr:FROM_NAME"/>}</xsl:if>
<xsl:if test="xdr:EMP_ID">{empid <xsl:value-of select="xdr:EMP_ID"/>}</xsl:if>
<xsl:if test="xdr:FROM_COMPANY">{rti <xsl:value-of
select="xdr:FROM_COMPANY"/>}</xsl:if>
<xsl:if test="xdr:FROM_DEPARTMENT">{dept <xsl:value-of
select="xdr:FROM_DEPARTMENT"/>}</xsl:if>
<xsl:if test="xdr:FROM_PHONE">{csi <xsl:value-of select="xdr:FROM_PHONE"/>}</xsl:if>
<xsl:if test="xdr:RETURN_EMAIL">{email <xsl:value-of
select="xdr:RETURN_EMAIL"/>}</xsl:if>
<xsl:if test="xdr:BILLINFO1">{billing <xsl:value-of select="xdr:BILLINFO1"/>}</xsl:if>
<xsl:if test="xdr:BILLINFO2">{billing2 <xsl:value-of
```
<!--END EMAIL DESTINATIONS-->
<!--PRINT DESTINATIONS-->
<xsl:template match="xdr:PRINT">
  {{execute}}{{type scale <xsl:if test="xdr:COPIES"><xsl:value-of select="xdr:COPIES"/></xsl:if>}}
  <xsl:if test="xdr:PRINTER_NAME">{lp <xsl:value-of select="xdr:PRINTER_NAME"/>}</xsl:if>
  <xsl:if test="@unique_id">{unique_id <xsl:value-of select="@unique_id"/>}</xsl:if>
  <xsl:if test="xdr:INCLUDE_INC">{include <xsl:value-of select="xdr:INCLUDE_INC"/>}</xsl:if>
  <xsl:if test="xdr:TO_COMPANY">{company <xsl:value-of select="xdr:TO_COMPANY"/>}</xsl:if>
  <xsl:if test="xdr:TO_CONTACTNUM">{voice <xsl:value-of select="xdr:TO_CONTACTNUM"/>}</xsl:if>
  <xsl:if test="xdr:COVERSHEET">{cover <xsl:value-of select="xdr:COVERSHEET"/>}</xsl:if>
  <xsl:if test="xdr:TO_NAME">{contact <xsl:value-of select="xdr:TO_NAME"/>}</xsl:if>
  <xsl:if test="xdr:INCLUDE_DEF">{include <xsl:value-of select="xdr:INCLUDE_DEF"/>}</xsl:if>
</xsl:template>
<!--END PRINT DESTINATIONS-->
XML Submit Reply DTD

The XML Submit Reply document conforms to the following DTD:

```xml
<!ELEMENT XML_FAX_SUBMIT_REPLY (FAX+)>
<!ATTLIST FAX unique_id CDATA #REQUIRED>
<!ELEMENT FAX (STATUS_CODE, STATUS_MSG)>
<!ELEMENT STATUS_CODE (#PCDATA)>
<!ELEMENT STATUS_MSG (#PCDATA)>
```
XML Query Schema

<?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>
Querying RightFax Using an XML Document

To query the system using XML you must create a document which conforms to the XML_FAX_QUERY_schema specified.

Now that you have successfully sent a fax using XML you have decided to test querying the system using an XML document. The XML document you would create might look like this:

```xml
<?xml version="1.0" ?>
<XML_FAX_QUERY
 xmlns="d:\RightFax\production\xml\schemas\XML_FAX_QUERY_schema.xml">
 <QUERIES>
  <QUERY faxtype="BOTH">
   <UNIQUE_ID>PRODXML:0001</UNIQUE_ID>
   <DATE_RANGE start="2000-12-08T18:39:09-08:00"
                end="2001-12-10T18:39:09-08:00"/>
   <TO_FAXNUM>1111</TO_FAXNUM>
   <RF_USER>test</RF_USER>
   <STATUS>0</STATUS>
  </QUERY>
 </QUERIES>
</XML_FAX_QUERY>
```
To test this document:

1. Open XML Spy on your computer.
2. Create a new XML document.
3. Specify that you will associate the document with a schema.
4. Browse to the following directory:

   D:\RightFax\production\xml\schemas\XML_FAX_QUERY_schema.xml

5. Type in the XML document above as demonstrated by your instructor.
6. Save the XML document to D:\XMLSampleCode\QuerySample\ 
7. Name the document BasicQuery.XML
8. Copy BasicQuery.xml to the directory you specified for query documents in the previous section.
9. Where will you find the results of your query?
Common Elements of Query Schema

When creating a query XML document there are a number elements and attributes that are required or used on a regular basis. For your convenience we have listed these elements and attributes with a brief description in parent / child order:

**XML_FAX_QUERY** (root element)

**QUERIES** (Required parent element that specifies queries to be submitted to server)

**QUERY** (Required parent element that specifies query information)

  faxtype (Attribute that specifies the type of faxes to query for)

  **UNIQUE_ID** (Element that specifies the id of the fax to query for)

  **DATE_RANGE** (Empty element for a date range query)

    start (Attribute that specifies start date of search)

    end (Attribute that specifies end date of search)

  **TO_FAXNUM** (Element that specifies the fax number to query for)

  **RFUSER** (Element that specifies the RightFax user id to query for)

  **STATUS** (Element that specifies the status code to query for)
Using the XML_FAX_QUERY_schema

After your test has successfully returned results, the CIO requests that you attempt another test that adds more complexity to your document. To complete this, perform the following exercises:

1. Query the database for all faxes associated with a specified user.

2. Query the database for outbound faxes that were sent to a specific fax number.

3. Query the database for outbound faxes sent to a specific fax number by a specific user.

4. Query the database for inbound faxes sent to a specific user.
XML Query Reply DTD

The XML Query Reply document conforms to the following DTD:

```xml
<!ELEMENT XML_FAX_QUERY_REPLY (FAXSTATUS+)>
<!ELEMENT FAXSTATUS (STATUS_CODE, STATUS_MSG, DELETED?,
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 DELETED (#PCDATA)>
<!ELEMENT ERROR_CODE (#PCDATA)>
<!ELEMENT DISPOSITION (#PCDATA)>
<!ELEMENT TERMSTAT (#PCDATA)>
<!ELEMENT OWNER_ID (#PCDATA)>
<!ELEMENT TO_FAXNUM (#PCDATA)>
<!ELEMENT TO_CONTACTNUM (#PCDATA)>
<!ELEMENT TO_NAME (#PCDATA)>
<!ELEMENT TO_COMPANY (#PCDATA)>
<!ELEMENT TO_CITYSTATE (#PCDATA)>
<!ELEMENT FROM_NAME (#PCDATA)>
<!ELEMENT FROM_PHONENUM (#PCDATA)>
<!ELEMENT BILLINFO1 (#PCDATA)>
<!ELEMENT BILLINFO2 (#PCDATA)>
<!ELEMENT CREATE_DATETIME (#PCDATA)>
<!ELEMENT SENDTIME (#PCDATA)>
<!ELEMENT REMOTEID (#PCDATA)>
<!ELEMENT SEND_DATETIME (#PCDATA)>
<!ELEMENT SEND_CHANNEL (#PCDATA)>
<!ELEMENT CUSTOM1 (#PCDATA)>
```
XML Action Schema

```xml
<?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>
</Schema>
```
Taking Action on a Fax Using XML

To take action on a fax using XML you must create a document which conforms to the XML_FAX_ACTION_schema specified.

Now that you have successfully queried the system using XML you have decided to test taking action on a fax using an XML document. The XML document you would create might look like this:

```xml
<?xml version="1.0" ?>
<XML_FAX_ACTION
xmlns="d:\RightFax\production\xml\schemas\XML_FAX_ACTION_schema.xml">
  <FAX unique_id="PRODXML:0001">
    <FORWARD>
      <FAX_RECIPIENT>
        <TO_NAME>Fred Flintstone</TO_NAME>
        <TO_COMPANY>Acme, Inc.</TO_COMPANY>
        <TO_CONTACTNUM>555-6543</TO_CONTACTNUM>
        <COVERSHEET>auto.cov</COVERSHEET>
        <TO_FAXNUM>1112</TO_FAXNUM>
      </FAX_RECIPIENT>
    </FORWARD>
  </FAX>
</XML_FAX_ACTION>
```

1. Type in the XML document above as demonstrated by your instructor.

2. Save the XML document to D:\XMLSampleCode\ActionSample\ 

3. Name the document BasicAction.XML

4. Copy BasicAction.xml to the directory you specified for submit documents in the previous section.
Common Elements of Action Schema

When creating an action XML document there are a number elements and attributes that are required or used on a regular basis. For your convenience we have listed these elements and attributes with a brief description in parent / child order:

**XML_FAX_QUERY** (root element)

- **FAX** (Required parent element that specifies the fax to take action on)
  - **unique_id** (Attribute that specifies the unique id to take action on)
- **DELETE** (Element that specifies to delete the fax)
- **FORWARD** (Parent element that forwards the fax)
  - **FAX_RECIPIENT** (Parent element that specifies the fax recipient(s) of the forwarded faxes)
    - **TO_NAME** (Element that specifies the recipients’ name on the forwarded fax)
  - **COVERSHEET** (Element that specifies coversheet)
  - **TO_FAXNUM** (Required element that specifies the fax number)
- **CREATE_LIB_DOC** (Parent element that creates a library document from the fax)
  - **ID** (Element that specifies the library document id)
  - **DESCRIPTION** (Element that specifies the description of the library document)
Using the XML_FAX_ACTION schema

After your test has successfully forwarded the fax to TEST2, the CIO requests that you attempt another test that adds more complexity to your document. To complete this, perform the following exercises:

1. Forward a fax to multiple users.

2. Delete an outbound fax.

3. Create a library attachment.
XML Action Reply DTD

The XML Action Reply document conforms to the following DTD:

```xml
<!ELEMENT XML_FAX_ACTION_REPLY (ACTION_STATUS+)>  
<!ELEMENT ACTION_STATUS (STATUS_CODE, STATUS_MSG)>  
<!ATTLIST ACTION_STATUS unique_id CDATA #REQUIRED>  
<!ATTLIST ACTION_STATUS type (forward|delete) "delete">  
<!ELEMENT STATUS_CODE (#PCDATA)>  
<!ELEMENT STATUS_MSG (#PCDATA)>  
```
Receiving Notification Through XML

Notification from an XML Submit document utilizes the same output devices a standard FCL document would use. RightFax includes a sample XML Notification template which can be called using the NOTIFY_HOST element. To test notification through XML:

1. Create an output device in the production control panel applet and name it XMLNotify.
2. Enter the following into the command line:
   
   COPY $$ d:\xmlnotify

3. In XML Spy open an XML Fax Submit document that you saved from one of the previous exercises.
4. Add the NOTIFY_HOST element as a child of the FAX element.
5. Set the SuccessTemplate attribute to XML_FAX_NOTIFICATION.
6. Set the FailureTemplate attribute to XML_FAX_NOTIFICATION.
7. Set the Name attribute to XMLNotify.
8. Save the XML document to D:\XMLSampleCode\SubmitSample\Notification.XML.
9. Name the document Notification.XML.
10. Copy Notification.xml to the directory you specified for submit documents in the previous section.
11. Where is the notification document placed?
12. Open the document in Notepad.
Section 2
Java Interface to XML Fax

Objectives

Upon completion of this section, you should be able to:

• Submit a fax to be sent via a Java application you have created.
• Query the RightFax server via a Java application you have created.
• Take action on a fax via a Java application you have created.

Overview

The RightFax Java Interface to XML-Fax allows you to submit an outbound fax, query the RightFax server for the status of a fax, and perform actions such as forward or delete. It provides you with the tools to accomplish all of these tasks through Java.
Java Interface to XML Fax

Requirements

Before installing the Java Interface to XML-Fax interface you must have the following installed on your server:

1. RightFax Enterprise Server.
2. Production Module.
3. XML-Fax Module.
4. Microsoft Internet Information Server 4.0.
5. Microsoft Internet Explorer 5.0 or greater.

Installation

After completing the installation of the XML-Fax component you must install the Java-XML component from the JAVA Install folder which is created under the Production. This will install the documentation we will be referring to throughout the class into the RightFax\Production\xml\Java\Docs directory and will also create a RFJava_api zip file. To complete the installation:

1. Extract the .class files from the RFJava_api.zip file.
2. These files will be extracted to a RightFax directory. This directory’s path should be specified in the classpath environment variable of your operating system.
Submitting a Fax Through Java

To submit an outbound fax you will need to set sender info, recipient info, and add the content for the primary document (if any). These are encapsulated in an RFDoc object, in the RFaxSubmit object. Then you can add any attachments. The following code illustrates how to create a simple Java Fax Submit Application:

```java
import RightFax.*;
import java.net.MalformedURLException;
import java.net.UnknownHostException;
import java.io.IOException;
import java.util.*;

public class APIExample
{
    public static void main (String [] argv)
    {
        System.out.println ("Started");
        submitAFax ();
        System.out.println ("Ended");
    }

    private static void submitAFax ()
    {
        RFaxSubmit obfax = new RFaxSubmit();
        obfax.setTargetURL ("http://studentx/"");
        obfax.m_FaxDocument.setSenderInfo ("Administrator");
        try {
            obfax.m_FaxDocument.addRecipient ("555-7777");
        } catch (RFNoFaxNumberException nfne) {
            System.out.println ("No Fax Number Given");
            obfax.m_FaxDocument.setBody ("How about some body");
        } catch (RFInvalidIDException iide) {
            System.out.println (iide.toString());
        }
        obfax.m_FaxDocument.setBody ("How about some body");
        try {
            Vector obRetList = null;
            obRetList = obfax.submit ();
            if (obRetList != null) {
                int nSize = obRetList.size();
                for (int i = 0; i < nSize; i++)
                {
                    RFStatus obStat = (RFStatus)(obRetList.elementAt(i));
                    System.out.println ("ID: " + obStat.getID());
                    System.out.println ("Code: " + obStat.getStatusCode());
                    System.out.println ("Msg: " + obStat.getStatusMsg());
                }
            }
        }
    }
}
```

```java```
catch (UnknownHostException uhe) {
    System.out.println(uhe.toString());
  }
  catch (MalformedURLException mfue) {
    System.out.println(mfue.toString());
  }
  catch (IOException ioe) {
    System.out.println(ioe.toString());
  }
  catch (RFNoDataException nde) {
    System.out.println(nde.toString());
  }
If we look at the code section by section:

1. In this section we import all of the classes that will be utilized within the application.
2. On this line we create a public class named APIExample.
3. In this section we call the main method which begins the execution of the application. Within the method you:
   - Print to the screen that the application is started.
   - Call the SubmitAFax Method.
   - Print to the screen that the application has ended.
4. This line specifies the beginning of the SubmitAFaxMethod.
5. On this line we create a new RFaxSubmit Object and call it obfax.
6. On this line we specify the URL of the target server through the setTargetURL method of RFaxSubmit.
7. On this line we use the RFDOC object, called m_FaxDocument in RfaxSubmit, to specify the sender information. This is accomplished using the setSenderInfo method. All the parameters of this method are optional except the RightFax user id.
8. On this line we add recipient information through the addRecipientInfo method of m_FaxDocument. All the parameters of this method are optional except the fax number.
9. This section determines if any exception errors are thrown due to no fax number being given or an incorrect RightFax user id. If either of these exceptions are thrown, information is written to the screen.
10. On this line we add text to the body of the fax using the setBody method of m_FaxDocument.
11. On this line we vector the obRetList variable and set it to null.
12. On this line we send the fax using the submit method of RfaxSubmit and set obRetList equal to the returned values.
13. On this line we determine if there are return values in obRetList.
14. On this line we get the number of items in the list and pass it to the nSize variable. This way we know how many times to loop.
15. On this line we loop a number of times equal to nSize and use ‘i’ as the counter.

16. In this section we get the RFStatus object at the given position (i) in the list and print the results to the screen via the println method. Each status object holds the reply data for 1 destination.

17. This section determines if any exception errors are thrown due to an unknown host being given, a malformed URL, an IO error, or no RightFax data being passed. If any of these exceptions are thrown, information is written to the screen.
RFaxSubmit Class

The RFaxSubmit class is used to submit faxes to the server to be sent to recipient(s). The commonly used method associated with this class is addAttachment. This method allows you to add a TIF attachment to your document by supplying the filename and path as a parameter of the method. If you wish to add multiple attachments to the document you can call this method multiple times. An example of how to use the addAttachment method might be:

```java
obfax.addAttachment("d:/attachment/terms.tif");
obfax.addAttachment("d:/attachment/terms2.tif");
```

In the example obFax is an object variable that represents the RFaxSubmit class.

The m_FaxDocument field is also associated with the RFaxSubmit class and is used to reference the RFDoc class. This field will be used in conjunction with the methods associated with the RFDoc class to build the fax document. The addRecipient method is required to build the fax document and is associated with the RFDoc class. An example of how to use the addRecipient method might be:

```java
obfax.m_FaxDocument.addRecipient("520-321-2200")
```

In the example obFax is an object variable that represents the RFaxSubmit class, m_FaxDocument is the field associated with the RFDoc class, and “520-321-2200” is the fax number parameter being passed to the addRecipient method.

The addRecipient method is an overloaded method. An overloaded method allows the application designer to use the same method name multiple times but change the number of parameters being passed. In the example of addRecipient there are three possible ways you can use the method. To see the parameters that can be used for each method reference the index html doc.
## RFaxSubmit and RFDoc Classes

Commonly Used Classes and Methods for Sending a Fax

### Class RFaxSubmit (used to submit document to server)

<table>
<thead>
<tr>
<th>Method</th>
<th>Overloaded</th>
<th>Required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>addAttachment</td>
<td></td>
<td></td>
<td>Used to add attachments to fax documents</td>
</tr>
<tr>
<td>m_FaxDocument</td>
<td></td>
<td></td>
<td>Used to reference the RFDoc class</td>
</tr>
</tbody>
</table>

### Class RFDoc (used to build the fax document to be sent)

<table>
<thead>
<tr>
<th>Method</th>
<th>Overloaded</th>
<th>Required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>addRecipient</td>
<td>✔️</td>
<td>✔️</td>
<td>Used to specify the parameters of recipient</td>
</tr>
<tr>
<td>AddRecipient_email</td>
<td>✔️</td>
<td></td>
<td>Used to specify the parameters of an email recipient</td>
</tr>
<tr>
<td>AddRecipient_printer</td>
<td>✔️</td>
<td></td>
<td>Used to specify the parameters of a recipient printer</td>
</tr>
<tr>
<td>setSenderInfo</td>
<td>✔️</td>
<td>✔️</td>
<td>Used to specify the setting of the fax sender</td>
</tr>
<tr>
<td>SetBeginInclude</td>
<td></td>
<td></td>
<td>Used to specify an include file to place at beginning of document</td>
</tr>
<tr>
<td>setBody</td>
<td>✔️</td>
<td></td>
<td>Used to specify the body of the fax</td>
</tr>
<tr>
<td>setCoverText</td>
<td></td>
<td></td>
<td>Used to specify the notes on the coversheet</td>
</tr>
<tr>
<td>setDate</td>
<td></td>
<td></td>
<td>Used to delay the fax</td>
</tr>
<tr>
<td>setFormXCoord</td>
<td></td>
<td></td>
<td>Used to specify the x coordinate when specifying a form</td>
</tr>
<tr>
<td>setFormYCoord</td>
<td></td>
<td></td>
<td>Used to specify the y coordinate when specifying a form</td>
</tr>
<tr>
<td>setForm</td>
<td></td>
<td></td>
<td>Used to specify form placed on the document</td>
</tr>
</tbody>
</table>
Test the system by developing the following applications:

- Develop an application that sends a fax to your RightFax account from your RightFax account.

- Add the FCS.PCL coversheet to your fax. Be sure the “to” and “from” information will be displayed. Add some notes to the coversheet and send the fax to yourself.

- Create a begin include file with formatting information for your fax. Add the method to your application to utilize the begin include file you have created.

- Create three new users in EFM. Give each user a unique routing code. Send a fax to the three users you created in EFM.

- Add the Terms.tif to your document as an attachment.

- Send the document to yourself as an email. Your email address is studentXX@captraining.com. Open outlook and logon using the Outlook profile. Check to see if the email was delivered.
Querying the Fax Server

To perform a query on the status of a fax (or group of faxes), you will need to set the criteria for each query you want to do. The following code illustrates how to create a simple Java Fax Submit Application:

```java
import RightFax.*;
import java.net.MalformedURLException;
import java.net.UnknownHostException;
import java.io.IOException;
import java.util.*;

public class APIExample
{
    public static void main (String [] argv)
    {
        System.out.println ("Started");
        queryAFax ();
        System.out.println ("Ended");
    }

    private static void queryAFax ()
    {
        RFaxQuery obQfax = new RFaxQuery();
        obQfax.setTargetURL ("http://studentx/");
        try {
            obQfax.addQuery ("uniqueid");
            Vector obRetList = null;
            try
            {
                obRetList = obQfax.submit ();
                if (obRetList != null) {
                    int nSize = obRetList.size();
                    for (int i = 0; i < nSize; i++)
                    {
                        RFStatus obStat =
                        (RFStatus)(obRetList.elementAt(i));
                        System.out.println ("ID: " + obStat.getID());
                        System.out.println ("Code: " + obStat.getStatusCode());
                        System.out.println ("Msg: " + obStat.getStatusMsg());
                    }
                }
            }
            catch (UnknownHostException uhe) {
                System.out.println (uhe.toString());
            }
            catch (MalformedURLException mfue) {
                System.out.println (mfue.toString());
            }
            catch (IOException ioe) {
                System.out.println (ioe.toString());
            }
            catch (RFEmptyQueryException ege) {
                System.out.println (ege.toString());
            }
            catch (RFInvalidIDException iide) {
                System.out.println (iide.toString());
            }
            catch (RFNoDataException nde) {
                System.out.println (nde.toString());
            }
        }
        catch (RFInvalidQueryException iqe) {
            System.out.println (iqe.toString());
        }
    }
}
```
If we look at the code section by section:

1. In this section we import all of the classes that will be utilized within the application.

2. On this line we create a public class named APIExample.

3. In this section we call the main method which begins the execution of the application. Within the method you:
   - Print to the screen that the application is started.
   - Call the queryAFax Method.
   - Print to the screen that the application has ended.

4. This line specifies the beginning of the queryAFax Method.

5. On this line we create a new RFaxQuery Object and call it obQFax.

6. On this line we specify the URL of the target server through the setTargetURL method of RFaxQuery.

7. On this line we create the query through the addQuery method of the RfaxQueryObject. In the example we provide the unique id of the fax for the parameter.

8. This section determines if any exception errors are thrown due to no query parameters being given or an incorrect RightFax user id. If either of these exceptions are thrown, information is written to the screen.

9. On this line we vector the obRetList variable and set it to null.

10. On this line we submit the query to the RightFax server using the submit method of RFaxQuery and set obRetList equal to the returned values.

11. On this line we determine if there are return values in obRetList.

12. On this line we get the number of items in the list and pass it to the nSize variable. This way we know how many times to loop.

13. On this line we loop a number of times equal to nSize and use 'i' as the counter.

14. In this section we get the RFStatus object at the given position (i) in the list and print the results to the screen via the println method. Each status object holds the reply data for 1 destination.

15. This section determines if any exception errors are thrown due to an unknown host being given, a malformed URL, an IO error, or no RightFax data being passed. If any of these exceptions are thrown, information is written to the screen.
RFaxQuery Class

The RFaxQuery class is used to submit queries to the RightFax server. The results will be returned via a vector using the RFStatus class. The common methods used for the RFaxQuery class are:

Class RFaxQuery (used to query the server)

<table>
<thead>
<tr>
<th>Method</th>
<th>Overloaded</th>
<th>Required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>addQuery</td>
<td>✓</td>
<td></td>
<td>Used to query the server based on a specific time range</td>
</tr>
</tbody>
</table>
The CIO of CAF has asked you to test the system by developing the following applications:

- Create an application where you query the server for all outbound faxes sent by your account.

- Create an application where you query the server for all outbound faxes sent to a specific fax number.
Taking Action Against A Fax

To perform an action on a fax, you will need the unique_id for that fax. Once you have that, you can either delete the fax, or forward it to another fax number. If you don’t have the unique_id, you can obtain it by doing a query on the information that you do have (see above). The following code illustrates how to create a simple Java Fax Action Application:

```java
import RightFax.*;
import java.net.MalformedURLException;
import java.net.UnknownHostException;
import java.io.IOException;
import java.util.*;

public class APIExample
{
    public static void main (String [] argv)
    {
        System.out.println ("Started");
        actionOnAFax ();
        System.out.println ("Ended");
    }

    private static void actionOnAFax ()
    {
        RFaxAction obAfax = new RFaxAction();
        obAfax.setTargetURL ("http://studentx");
        try {
            obAfax.addDeleteAction ("uniqueid");
        } catch (RFNoIDException nie) {
            System.out.println (nie.toString());
        } catch (RFInvalidOpException iope) {
            System.out.println (iope.toString());
        }
        try {
            Vector obRetList = null;
            obRetList = obAfax.submit ();
            if (obRetList != null) {
                int nSize = obRetList.size();
                for (int i = 0; i < nSize; i++)
                {
                    RFStatus obStat = (RFStatus)(obRetList.elementAt(i));
                    System.out.println ("ID: " + obStat.getID());
                    System.out.println ("Code: " + obStat.getStatusCode());
                    System.out.println ("Msg: " + obStat.getStatusMsg());
                }
            }
        } catch (UnknownHostException uhe) {
            System.out.println (uhe.toString());
        } catch (MalformedURLException mfue) {
            System.out.println (mfue.toString());
        } catch (IOException ioe) {
            System.out.println (ioe.toString());
        } catch (RFNoDataException nde) {
            System.out.println (nde.toString());
        }
    }
}
```

1. import RightFax.*;
2. import java.net.MalformedURLException;
3. import java.net.UnknownHostException;
4. import java.io.IOException;
5. import java.util.*;
6. public class APIExample
7. {
8.     public static void main (String [] argv)
9.     {
10.        System.out.println ("Started");
11.        actionOnAFax ();
12.        System.out.println ("Ended");
13.    }
14.    private static void actionOnAFax ()
15.    {
16.        RFaxAction obAfax = new RFaxAction();
17.        obAfax.setTargetURL ("http://studentx");
18.        try {
19.            obAfax.addDeleteAction ("uniqueid");
20.        } catch (RFNoIDException nie) {
21.            System.out.println (nie.toString());
22.        } catch (RFInvalidOpException iope) {
23.            System.out.println (iope.toString());
24.        }
25.        try {
26.            Vector obRetList = null;
27.            obRetList = obAfax.submit ();
28.            if (obRetList != null) {
29.                int nSize = obRetList.size();
30.                for (int i = 0; i < nSize; i++)
31.                {
32.                    RFStatus obStat = (RFStatus)(obRetList.elementAt(i));
33.                    System.out.println ("ID: " + obStat.getID());
34.                    System.out.println ("Code: " + obStat.getStatusCode());
35.                    System.out.println ("Msg: " + obStat.getStatusMsg());
36.                }
37.            }
38.        } catch (UnknownHostException uhe) {
39.            System.out.println (uhe.toString());
40.        } catch (MalformedURLException mfue) {
41.            System.out.println (mfue.toString());
42.        } catch (IOException ioe) {
43.            System.out.println (ioe.toString());
44.        } catch (RFNoDataException nde) {
45.            System.out.println (nde.toString());
46.        }
47.    }
48.}
```
If we look at the code section by section:

1. In this section we import all of the classes that will be utilized within the application.
2. On this line we create a public class named APIExample.
3. In this section we call the main method which begins the execution of the application. Within the method you:
   - Print to the screen that the application is started.
   - Call the actionOnAFax Method.
   - Print to the screen that the application has ended.
4. This line specifies the beginning of the actionOnAFax Method.
5. On this line we create a new RFaxAction Object and call it obAfax.
6. On this line we specify the URL of the target server through the setTargetURL method of RFaxAction.
7. On this line we create the delete action through the addDeleteAction method of RfaxAction. In this example we use the faxes unique id as the parameter to the method.
8. This section determines if any exception errors are thrown due to an incorrect RightFax user id being given or when an operation is not set or is unauthorized. If either of these exceptions are thrown, information is written to the screen.
9. On this line we vector the obRetList variable and set it to null.
10. On this line we submit the delete action to the RightFax server using the submit method of RFaxAction and set obRetList equal to the returned values.
11. On this line we determine if there are return values in obRetList.
12. On this line we get the number of items in the list and pass it to the nSize variable. This way we know how many times to loop.
13. On this line we loop a number of times equal to nSize and use 'i' as the counter.
14. In this section we get the RFStatus object at the given position (i) in the list and print the results to the screen via the println method. Each status object holds the reply data for 1 destination.
15. This section determines if any exception errors are thrown due to an unknown host being given, a malformed URL, an IO error, or no RightFax data being passed. If any of these exceptions are thrown, information is written to the screen.
RFaxAction Class

The RFaxAction class is used to take action on a fax (specified by its’ unique id) on the server. The actions that you can take are, delete, forward to new fax number, and create a library document. The methods that are available for RFaxAction are:

Class RFaxAction (used to take action on a fax)

<table>
<thead>
<tr>
<th>Method</th>
<th>Overloaded</th>
<th>Required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>addDeleteAction</td>
<td></td>
<td></td>
<td>Used to delete the fax specified by the unique id</td>
</tr>
<tr>
<td>addForwardAction</td>
<td>✓</td>
<td></td>
<td>Used to forward the fax specified to a new number</td>
</tr>
<tr>
<td>addLibDocAction</td>
<td></td>
<td></td>
<td>Used to create a library document from the fax specified</td>
</tr>
</tbody>
</table>
Exercise

The CIO of CAF has asked you to test the system by developing the following applications:

- Create an application that forwards a fax to multiple users.
- Create an application that creates a library document from a specific fax.
Section 3
Understanding the COM Module

Objectives

Upon completion of this section, you should be able to:

• Setup the RightFax COM API module.
• Explain what properties, methods and events are.
• Explain the object hierarchy.
Overview

The RightFax COM object is contained in a single dynamic link library (DLL) file named Rfcomapi.dll. This file is installed as part of the RightFax client installation from the development server.

To install and register the COM object DLL:

1. On the RightFax COM development machine, open a view of the \RightFax\Client\Win32 folder on the RightFax development server.
2. Run Setup.exe.
3. Click Next at the introductory screen.
4. Enter a user name and organization name. If you are installing on a Windows 2000 machine, you will also have an option to install for the current user only or all user profiles on the machine. (To install for all users you must be an administrator.) Click Next.
5. Select the Complete install option.
6. Follow the prompts to complete the installation.

Note: For detailed information on installing the RightFax client, refer to the RightFax Administrator’s Guide.

The COM object file Rfcomapi.dll is registered and installed in the folder you specified for the client software during the install. The files RFWin32.dll, Rfi32rpc.ndr, and Rfi32smb.ndr (RightFax API files upon which the COM DLL is dependent) are also copied to this folder.

The RightFax COM API

The RightFax COM API is an interface to the standard Component Object Model (COM) services included in all 32-bit Windows operating systems. Like the standard RightFax API, the COM API consists of tools that allow you to customize your RightFax products. In addition, there are a variety of pre-defined objects stored in the RightFax COM API library (RFCOMAPILib).

Understanding Objects, Properties, and Methods

The RightFax COM API library contains objects, which are discrete entities within the RightFax system that can be accessed and manipulated. Users, attachments, and library documents are examples of objects. Multiple objects of the same type can be grouped together in objects called Collections. For example, the object called Faxes is a collection of Fax objects.
Each object has properties, which are descriptive aspects of that object. For example, UserID and Password are properties of the User object. Properties may be used to identify an object, or to set permissions for the object, or to show status.

Each object also has methods, which are actions that can be performed on an object. Sending a fax, adding an object to a collection, and logging on to a server are examples of methods.

**The FaxServer Object**

Although many objects share similar properties and methods, one object is unique. The FaxServer object must be created before you can access any of the other COM API components. It is also the only object that can be used to create objects within the RightFax COM API.

The FaxServer object is also an important part of the Event Handler. You must define a server object in order to create applications that perform tasks based on events (such as sending or receiving a fax).

**Enumerated Data Types**

Some of the properties in the RightFax COM API library require specific, predetermined values. These values are included in the library as enumerated data types. The data type generally has both a text name and a number that represents that text. For example, the FaxStatus property of the Fax object must contain one of the valid values enumerated in the FaxStatusType data type (such as fsSend, fsInPrint, or fsQueuedforOCR). The enumerated values are defined in this guide with the property to which they apply.

**The Object Hierarchy**

Objects in the RightFax COM Object are organized hierarchically. At the top of this hierarchy is the FaxServer object, which contains all the other objects in the object model. The Chart on the next page is a visual representation of the RightFax COM object hierarchy:
Figure 3.1 The RightFax COM Object Hierarchy
Exploring the COM Object Library Using the Visual Basic Object Browser

The RightFax COM API library uses descriptive interface definition language (IDL). This means that every function has some documentation on what it is supposed to do. To access this information, open the Object Browser in Visual Basic.

To open the Object Browser:

1. Select Project > Object Browser from the menu, or press F2, or select Object Browser from the toolbar. This opens the Object Browser dialog box.

2. In the object list box, select RFCOMAPILib.

   You can now explore the entire object model. Under Classes is a list of all the objects and enumerations in the COM API. Under Members is a list of the methods, properties, constants, and events of the selected object.

3. In the Classes list, click FaxServer.

4. In the Members of FaxServer list, click Users. A description of the User object appears in the bottom of the dialog box.
Exercise

1. Open the object browser and explore the RightFax COM API objects.
   - List three properties associated with the FAX object that are read only.

   - What methods are associated with all of the collections.

   - List two enumerations.
Section 4
Creating a Simple Application

Objectives

Upon completion of this section, you should be able to:

- List the steps required to design a simple application.
- Create and log on to a FaxServer object.
- Compile an application.
Steps to Design a Simple Application

The RightFax COM Module consists of a variety of objects you can use to create any type of application your company needs. This section describes the steps required to build a simple project using Visual Basic. In this project, you’ll create a window that displays a list of users. When you click a user’s name, a list of the faxes for that user will appear.

To build this project, complete the following steps:

Step 1: Start a New Visual Basic Project
Step 2: Set Up the Project Form
Step 3: Create and Log On to a New Fax Server Object
Step 4: Display the Users in the List Box
Step 5: List Faxes for the Selected User
Step 6: Add an Exit Button
Step 7: Test and Compile the Program

Step 1 - Start a New Visual Basic Project

When you first start Visual Basic, you’ll need to create a new project. You’ll also need to ensure that the RightFax COM components are available to the Visual Basic application.

1. Run Visual Basic.
2. Choose the New Standard EXE Project icon.
3. On the main menu, select Project > References.
4. In the Available References list, click RightFax COM API 1.0 Type Library.
Step 2 - Set Up the Project Form

This tutorial will create a program that lists the users in a window. You must define the appearance of the window by defining the project Form properties. In this step, you’ll add two list boxes to the form, one to display the users and the other to display the list of faxes for the selected user.

1. If it is not already displayed, turn on the properties window. On the main menu, select View > Properties Window.
2. Rename the Form window by changing the Name property to frmMain. It is usually a good idea to preface your names with the type of object (such as frm for form or str for string).
3. Title the window by changing the Caption property to Users.
4. Create a list box. Double-click the list box icon on the right side of the Visual Basic project window.
5. Resize the list box by clicking and dragging the handles on the sides of the box.
Creating a Simple Application

6. Rename the list box by changing the Name property to lst_Users (where lst represents a list box).

7. Add another list box, and resize as necessary.

8. Name the second list box lst_FaxList.

Step 3 - Create and Log On to a New Fax Server Object

In order to access the COM API objects, you must first create a new server object that will be referenced throughout your program. Place this code in the main form and make declarations using the Public statement so that the RightFax server will be available to other segments of code. You’ll also need to specify the information required to log on to the server.

1. If not already displayed, display on the code window by selecting View > Code from the main menu.

2. In the General section, create a global variable called gl_MyFaxServer that is of the type RFCOMAPILib.FaxServer. The gl indicates it is a global variable and is available to all forms in the project.

   Public gl_MyFaxServer As RFCOMAPILib.FaxServer
Note that as you enter letters, VB displays a list of available options. For example, after you’ve typed RFC, the full name RFCOMAPILib will be highlighted. Type a period to display the list of objects in this library. As you enter the first letters of the object name, available options will be highlighted. When the one you want, FaxServer, is highlighted, press ENTER to select it. This creates the variable MyFaxServer (although it is still empty at this point).

3. Now that you’ve created the variable, set it to contain the objects in the RFCOMAPILib object. From the object list, choose Form. The procedure list should display the text Load, indicating that the code you enter will be executed when the form is loaded. Type the following between the existing Sub and End Sub statements to set the variable created in step 2.

```vba
Set gl_MyFaxServer = New RFCOMAPILib.FaxServer
```

![Figure 4.3 Setting the Variable](image)
4. Specify the login information that will be required to access the server. If you are using NT authorization, you may set the UseNTAuthentication property to TRUE, and ignore the AuthorizationUserID and AuthorizationUserPassword properties. Type this code between the predefined Sub and End Sub statements.

```vbnet
gl_MyFaxServer.ServerName = "servername"
gl_MyFaxServer.UseNTAuthentication = False
gl_MyFaxServer.AuthorizationUserID = "admin"
gl_MyFaxServer.AuthorizationUserPassword = "secretword"
gl_MyFaxServer.Protocol = cpNamedPipes
```

**Step 4 - Display the Users in the List Box**

To display a list of users in the list box you’ve named lstUsers, you’ll need to specify the properties of the User object that you want to display. This procedure also describes creating labels for the list box.

1. Display the form window. Select View > Object from the main menu.
2. Click the label icon, and then drag the cursor to draw a box where you want the first label to be (above the list box, aligned with the left side of the box).
3. Change the Caption property for the label to **Users**.
4. Click the label icon again, and then draw a box above and on the right side of the list box.
5. Change the Caption property for this label to **User ID**.

![Figure 4.4 Creating Labels](image_url)
6. Return to the **Code** window.

7. Declare the variables you are going to need. These variables are not global, so use the “Dim” statement instead of “Public.”
   
   ```vbnet
   Dim str_UserName As String
   Dim str_UserID As String
   ```
   
   This creates the two variables, but they are still empty.

8. Next, store the fields from the user object in the two new variables by entering the appropriate code. From the object list, choose **Form**. The procedure list should display the text **Load**, indicating that the code you enter will be executed when the form is loaded.

9. Enter the following code after the authentication statements but before the End Sub statement. Note that the first two lines are comments because they are preceded by an apostrophe (').

   ```vbnet
   'Display the list of user names and IDS in the lst_Users box.
   For Each User in gl_MyFaxServer.Users
       str_UserID = User.ID
       str_UserName = User.UserName
       lst_Users.AddItem str_UserID + vbTab + str_UserName
   Next
   ```
   
   The `vbTab` entry aligns the two columns.

**Step 5 - List Faxes for the Selected User**

In this sample application, a click on a user name will get a list of the user’s faxes. In this step, we’ll store the selected user to the variable and create a button that fills in the list box we created earlier.

1. From the object list on the left side of the screen, choose the name of the second list box we created earlier (**lst_FaxList**).

2. Enter the following code between the Sub and End Sub statements.

   ```vbnet
   Dim objUser as RFCOMAPILib.User
   Set objUser = gl_MyFaxServer.Users(lst_Users.ListIndex+1)
   gl_SelectedUser = objUser.ID
   ```

3. Click the **Command Button** icon to add a button beneath the list boxes. Set the button Name property to **btn_ListFaxes** and the Caption property to **List Faxes**.
4. In the **Code** window **Declarations** section, declare a global variable that will store the ID of the selected user. Make it a global variable so that you can use it in other forms, and place it with the other global variables you’ve already entered.

```vbnet
Public gl_SelectedUser As String
```

5. You’ll also need to declare variables for the information you want to appear in the Fax List. While you could declare each variable individually as when declaring the user variables, you can also combine the variables into a single statement.

```vbnet
Dim obj_Fax as RFCOMAPILib.Fax
Dim str_File As String, str_FromName As String
```

6. From the object list, choose the name of the second button we created earlier (**btn_ListFaxes**).

7. Enter the following code between the predefined Sub and End Sub statements.

```vbnet
For Each obj_Fax in MyFaxServer.Faxes(gl_SelectedUser)
    str_FaxFileName = obj_Fax.FaxFilename
    str_FromName = obj_Fax.FromName
    lst_FaxList.AddItem str_FaxFileName + vbTab + str_FromName
Next
```
Step 6 - Add an Exit Button

You want to be able to shut down your program, so you’ll need to add one more button.

1. In the Object window, select the Command Button icon.
2. Change the name of the button to btn_Exit and the caption to Exit.
3. In the Code window, choose btn_Exit from the procedure list.
4. Enter the following code between the Sub and End Sub entries.
   
   Unload frmMain

Step 7 - Test and Compile the Program

At any time while creating your project, you can press F5 to run a test of the code so far. You may see a lot of error messages if you haven’t completed all of the steps. If you’ve followed all of the steps in this chapter, a list of users will appear. When you click a user, a list of faxes will appear if that user has any faxes in the system.
When everything works the way you want it to, save and compile the program by following these steps.

1. Press F5 to check the project and make sure it works as expected.
2. Select File > Save. A message appears asking for a name for the form you’ve created.
3. Enter a name. You can use either ListUsers or UserFaxes or some other name with meaning to you. Click Save.
4. A message appears asking for a name for the whole project. You can have multiple forms in a single project. Enter a name such as Tutorial, and then click Save.
5. Select File > Make Tutorial.exe. If you’ve entered a name other than “Tutorial,” that name will appear in the menu.
   
   Note: This will not work if you have errors, so make sure you test the program first.

6. Enter a name for the executable file. You can use the name of the project (in this case, “Tutorial”) or you can change it. Click OK. The resulting file, Tutorial.exe, can be run on any system that is running the RightFax software with the COM module.
Exercise

1. Create the sample application from the tutorial.
Creating a Simple Application
Section 5
The FaxServer and ServerInfo Objects

Objectives

Upon completion of this section, you should be able to:

- Initiate a new FaxServer object.
- Describe the properties, methods and events associated with FaxServer.
- Describe the properties, methods and events associated with ServerInfo.
The FaxServer and ServerInfo Objects

FaxServer Object

The FaxServer object is the base object in the RightFax COM Module and must be created before you can work with or create any other objects. This is the only object that can be created externally. This section describes all of the methods, properties, and events associated with the FaxServer.

The ServerInfo object contains information that describes the FaxServer in a read-only format. This section describes all of the ServerInfo properties.

Note: All examples were created and verified using Microsoft Visual Basic 6.0. If you are using a different development environment, some of the examples in this section may not work as expected.

FaxServer

If you do not specify one of the following properties when working with the FaxServer object, value you enter will be used as the ServerName.

Properties

AuthorizationUser
Returns the entire User object of the authorized User. Note that the User must first be logged on to the server before accessing this property.

AuthorizationUserID
Specifies the UserID that will be used to log on to the server. You must also specify the AuthorizationUserPassword property. This is not necessary if UseNTAuthentication is set to True.

AuthorizationUserPassword
Specifies the password of the user specified in AuthorizationUserID property.

BillingCodes
Returns the BillingCodes collection.
**CoverSheets**
Returns the CoverSheets collection.

**CreateObject**
This property creates a new object of the specified CreateObjectType. For example, to create and name a folder object, you might use this:

```vba
Set obFolder = MyFaxServer.CreateObject(coFolder)
obFolder.ID = "foldername"
```

The object type must be one of the valid enumerated values, listed here with their numeric equivalents.

**Table 5.1. CreateObjectType Enumerated Values**

<table>
<thead>
<tr>
<th>Name</th>
<th>Numeric Value</th>
<th>See Also</th>
</tr>
</thead>
<tbody>
<tr>
<td>coAttachment</td>
<td>0</td>
<td>Attachment object</td>
</tr>
<tr>
<td>coBillingCode</td>
<td>1</td>
<td>BillingCode object</td>
</tr>
<tr>
<td>coCoverSheet</td>
<td>2</td>
<td>CoverSheet object</td>
</tr>
<tr>
<td>coDelegatee</td>
<td>3</td>
<td>Delegatee object</td>
</tr>
<tr>
<td>coDelegator</td>
<td>4</td>
<td>Delegator object</td>
</tr>
<tr>
<td>coFax</td>
<td>5</td>
<td>Fax object</td>
</tr>
<tr>
<td>coFolder</td>
<td>6</td>
<td>Folder object</td>
</tr>
<tr>
<td>coForm</td>
<td>7</td>
<td>Form object</td>
</tr>
<tr>
<td>coGroup</td>
<td>8</td>
<td>Group object</td>
</tr>
<tr>
<td>coLibraryDocument</td>
<td>9</td>
<td>LibraryDocument object</td>
</tr>
<tr>
<td>coPhoneItemGroup</td>
<td>10</td>
<td>PhoneItemGroup object</td>
</tr>
<tr>
<td>coPhoneItemElement</td>
<td>11</td>
<td>PhoneItemElement object</td>
</tr>
<tr>
<td>coPrinter</td>
<td>12</td>
<td>Printer object</td>
</tr>
<tr>
<td>coSignature</td>
<td>13</td>
<td>Signature object</td>
</tr>
<tr>
<td>coUsers</td>
<td>14</td>
<td>Users collection</td>
</tr>
</tbody>
</table>

**Events**
Returns the Events interface. You will need to store this interface to maintain connection to the events as they occur. For example:
The FaxServer and ServerInfo Objects

Private Sub Form_Load()
    Dim rfFaxAPI As RFCOMAPILib.FaxServer
    Dim WithEvents FaxEventHandler As RFCOMAPILib.FaxServer
    Dim rfEvents As RFCOMAPILib.Events
    Set rfFaxAPI = New FaxServer
    Set FaxEventHandler = rfFaxAPI
    Set rfEvents = rfFaxAPI.Events
End Sub

Fax
Returns a specific fax identified by either the FaxHandle or the unique fax ID string.

Faxes
Returns the Faxes collection for a specified user. You must specify the UserID of the Faxes collection that you want.

Forms
Returns the Forms collection.

Group
Returns a specific group identified by the unique GroupID string.

Groups
Returns the Groups collection.

ImageExtensionToPageNumber
RightFax stores each page of a fax as an image file with a unique file name. The extension of the file name indicates the page number. This property returns the appropriate page number of the fax (such as page 9) for the RightFax file name extension (such as .309) that you specify.

LibraryDocument
Returns a specific LibraryDocument identified by either the LibraryDocumentHandle or the unique LibraryDocumentID.

LibraryDocuments
Returns the LibraryDocuments collection.

PageNumberToImageExtension
RightFax stores each page of a fax as an image file with a unique filename. The extension of the filename indicates the page number. This property returns the RightFax filename extension (such as .309) for the page number (such as page 9) that you specify.
**PhoneBook**
Returns the PhoneBook collection for the specified user. If you do not specify a user by entering the UserID, this property will return a blank PhoneBook collection.

**Printer**
Returns a specific Printer identified by either the PrinterHandle or the unique PrinterID string.

**Printers**
Returns the Printers collection.

**Protocol**
Specifies the CommunicationProtocolType of the FaxServer. The specified protocol type must be one of the valid enumerated values, listed here with their numeric equivalents.

<table>
<thead>
<tr>
<th>Name</th>
<th>Numeric Value</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>cpNamedPipes</td>
<td>1</td>
<td>Use this value for automatic protocols</td>
</tr>
<tr>
<td>cpIPXOS2</td>
<td>2</td>
<td>None</td>
</tr>
<tr>
<td>cpSPX</td>
<td>3</td>
<td>None</td>
</tr>
<tr>
<td>cpTCPPIP</td>
<td>4</td>
<td>None</td>
</tr>
<tr>
<td>cpIPX</td>
<td>5</td>
<td>None</td>
</tr>
<tr>
<td>cpSecTCPPIP</td>
<td>6</td>
<td>None</td>
</tr>
<tr>
<td>cpSecSPX</td>
<td>7</td>
<td>None</td>
</tr>
</tbody>
</table>

**ServerInfo**
Returns the ServerInfo object.

**ServerName**
Specifies the name of the FaxServer object. This property is the default; if you do not name a property, RightFax will assume you are specifying a ServerName.

**Signatures**
Returns the Signatures collection.

**SupportInfo**
Returns a text file containing the support information for the server, such as the phone number for customer support.
The FaxServer and ServerInfo Objects

UseNTAuthentication
A Boolean property, this indicates whether or not to use Windows NT Authorization. The default is True or 1: use the currently logged in ID and password for authentication. If you do not use NT authorization, you must specify the AuthorizationUserID and AuthorizationUserPassword.

User
Returns a specific User identified by either the UserHandle or the unique UserID string.

Users
Returns the Users collection.

Version
Returns the version number of the RightFax COM API.

Methods

CloseServer
This method closes the communications connection to the fax server. This is normally done when the IFaxServer interface is released. To change the server manually, close this connection before changing the server name.

OpenServer
Opens a communications connection to the fax server. This is normally done automatically the first time you do something that requires a server connection.

Events

The FaxServer is the only RightFax object that has events. There are three basic steps to using the EventHandler:

1. Set the object variables that will handle the event during the initialization of the Form.
2. Declare one or more functions that will be called by the COM API when the event happens.
3. Set the corresponding variable in the objects that initiate the event. For example, to make use of the OnArchiveEvent, you would have to set the IsArchiveEnable property to True in the User object.

OnArchiveEvent
This event is generated when a fax needs to be archived. Set the IsArchiveEnable property in the User object to initiate an OnArchiveEvent.
The FaxServer and ServerInfo Objects

**OnCompleteEvent**
This event is generated when a fax is sent that has the GenerateCompletionEvent flag set to True or 1.

**OnMessageEvent**
This event is generated when a message needs to be displayed to the user (such as an error message).

**OnNewFaxEvent**
This event is generated when a new fax for a specific user comes in to the system.

**OnValidateEvent**
This event is generated when a user has the requires validations flag set to True or 1. The fax will stay in the user’s queue indefinitely until the fax is validated by checking for this event.

**Example**

The following code declares the object variables that are needed to create a FaxServer object and then logs on to the server.

' Declare the object variables that will be used in the application
Dim rfFaxAPI As RFCOMAPILib.FaxServer
Dim WithEvents FaxEventHandler As RFCOMAPILib.FaxServer
Dim rfEvents As RFCOMAPILib.Events
DIM MyFaxServer As RFCOMAPILib.FaxServer
Public Sub CreateGroup()
  Set MyFaxServer = New FaxServer
  ' Log on to the fax server
  MyFaxServer.ServerName = "FaxServer"
  MyFaxServer.Protocol = cpNamedPipes
  MyFaxServer.AuthorizationUserID = "studentX"
  MyFaxServer.UseNTAuthentication = True
End Sub

**ServerInfo**

ServerInfo provides detailed information about the server you are communicating with. To retrieve this object, use the ServerInfo property in the FaxServer object.

**Properties**

**BaseDirectoryPath**
Returns the path to the base directory.
BillingCodeDescription1
Returns the description for BillingCode1. This can be used as a field label for Billing Code1.

BillingCodeDescription2
Returns the description for BillingCode2. This can be used as a field label for Billing Code2.

BuildDate
Returns the date when the FaxServer was built. This is in hex format 0xYYYYMMDD.

DefaultANSICodePage
Page on the server that contains the ANSI codes.

DefaultOEMCodePage
Page on the server that contains the OEM codes.

ImageDirectoryLocation
Path to the image directory starting at the root RightFax directory.

ImageDirectoryPath
Path to the image directory starting at the system root directory (includes the RightFax directory path).

ImageDirectoryShareName
Returns the share name from the LANMAN/LANSERVER.

IsDatabaseANSI
A Boolean property, this indicates whether or not the fax server is running in ANSI mode. True or 1 indicates ANSI; False or 0 indicates OEM character support.

IsDocsOnDemandLicensed
A Boolean property, this indicates whether or not the server has a license for the RightFax Docs-on-Demand™ module. True or 1 indicates there is a license; False or 0 indicates there is not.

IsEnterprise
A Boolean property, this indicates whether or not the server has a license for RightFax Enterprise™ software. True or 1 indicates the server does have a license; False or 0 indicates the server does not have a license.

IsIPPlusConnectorLicensed
A Boolean property, this indicates whether or not the server has an IP Plus Connector license. True or 1 indicates the server does have a license; False or 0 indicates the server does not have a license.
IsOCRConverterLicensed
A Boolean property, this indicates whether or not the server has a RightFax OCR Converter™ software license. True or 1 indicates the server does have a license; False or 0 indicates the server does not have a license.

IsOCRRouterLicensed
A Boolean property, this indicates whether or not the server has a RightFax OCR Router™ software license. True or 1 indicates the server does have a license; False or 0 indicates the server does not have a license.

IsOEMCPModeEnabled
A Boolean property, this indicates whether or not the original equipment manufacturer code page mode is enabled. True or 1 indicates the mode is enabled; False or 0 indicates it is not.

IsPDFModuleLicensed
A Boolean property, this indicates whether or not the server has a license for the RightFax PDF module. True or 1 indicates the server does have a license; False or 0 indicates the server does not have a license.

IsProductionFilterLicensed
A Boolean property, this indicates whether or not the server has a license for the RightFax Filter for Production™ software. True or 1 indicates the server does have a license; False or 0 indicates the server does not have a license.

IsProductionINLLicensed
A Boolean property, this indicates whether or not the server has a license for the RightFax InternetLink™ module. True or 1 indicates the server does have a license; False or 0 indicates the server does not have a license.

IsProductionLicensed
A Boolean property, this indicates whether or not the server has a RightFax Integration Module™ license. True or 1 indicates the server does have a license; False or 0 indicates the server does not have a license.

IsProductionNotifierLicensed
A Boolean property, this indicates whether or not the server has a production notifier license. True or 1 indicates the server does have a license; False or 0 indicates the server does not have a license.

IsSatelliteLicensed
A Boolean property, this indicates whether or not the server has a RightFax Satellite™ server license. True or 1 indicates the server does have a license; False or 0 indicates the server does not have a license.
**The FaxServer and ServerInfo Objects**

**IsSecureDocsLicensed**  
A Boolean property, this returns True or 1 if the server has a SecureDocs license, and returns False or 0 if it does not.

**IsSerialValid**  
A Boolean property, this indicates whether or not the SerialNumber property contains a valid RightFax serial number. True or 1 indicates the number is valid; False or 0 indicates it is not.

**IsSmallBusinessServerLicensed**  
A Boolean property, this indicates whether or not the server has a Small Business Server license. True or 1 indicates the server does have a license; False or 0 indicates the server does not have a license.

**IsTeleconnectLicensed**  
A Boolean property, this indicates whether or not the server has a RightFax TeleConnect™ module license. True or 1 indicates the server does have a license; False or 0 indicates the server does not have a license.

**MaximumLicensedUsers**  
Returns the maximum number of users licensed to use the server. A zero indicates no maximum, which allows an unlimited number of users.

**NumberOfLicensedChannels**  
Returns the number of channels licensed for this server.

**RequiredFields**  
Indicates whether or not the specified object is required. To specify the object you want to check, send the RequiredFieldIndexType enumerated value from the following list.
Table 5.3. RequiredFieldIndexType Enumerated Values

<table>
<thead>
<tr>
<th>Name</th>
<th>Number equivalent</th>
<th>See Also</th>
</tr>
</thead>
<tbody>
<tr>
<td>rfiToName</td>
<td>0</td>
<td>ToName</td>
</tr>
<tr>
<td>rfiToFaxNumber</td>
<td>1</td>
<td>ToFaxNumber</td>
</tr>
<tr>
<td>rfiToVoiceNumber</td>
<td>2</td>
<td>ToVoiceNumber</td>
</tr>
<tr>
<td>rfiToCompany</td>
<td>3</td>
<td>ToCompany</td>
</tr>
<tr>
<td>rfiToCityState</td>
<td>4</td>
<td>ToCityState</td>
</tr>
<tr>
<td>rfiFromName</td>
<td>5</td>
<td>FromName</td>
</tr>
<tr>
<td>rfiFromPersonalVoiceNumber</td>
<td>6</td>
<td>FromPersonalVoiceNumber</td>
</tr>
<tr>
<td>rfiFromPersonalFaxNumber</td>
<td>7</td>
<td>FromPersonalFaxNumber</td>
</tr>
<tr>
<td>rfiFromGeneralVoiceNumber</td>
<td>8</td>
<td>FromGeneralVoiceNumber</td>
</tr>
<tr>
<td>rfiFromGeneralFaxNumber</td>
<td>9</td>
<td>FromGeneralFaxNumber</td>
</tr>
<tr>
<td>rfiBillCode1</td>
<td>10</td>
<td>BillCode1</td>
</tr>
<tr>
<td>rfiBillCode2</td>
<td>11</td>
<td>BillCode2</td>
</tr>
</tbody>
</table>

The return value will be one of the following RequiredFieldType enumerated values.

Table 5.4. RequiredFieldType Enumerated Values

<table>
<thead>
<tr>
<th>Name</th>
<th>Numeric equivalent</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>rfiNotRequired</td>
<td>0</td>
<td>The object is not required.</td>
</tr>
<tr>
<td>rfiSend</td>
<td>1</td>
<td>The object is only required when sending a fax.</td>
</tr>
<tr>
<td>rfiReceive</td>
<td>2</td>
<td>The object is only required when receiving a fax.</td>
</tr>
<tr>
<td>rfiAlways</td>
<td>3</td>
<td>The object is always required.</td>
</tr>
</tbody>
</table>

**SerialNumber**
Returns the server serial number. Use the IsSerialValid property to determine whether or not the serial number is valid.

**ServerName**
Returns the name of the server currently being used for communication.
ServerSpecial
Returns the special type of the server. The special type will be one of the following ServerSpecialType enumerated values.

Table 5.5. ServerSpecialType Enumerated Values

<table>
<thead>
<tr>
<th>Enumerated value</th>
<th>Numeric equivalent</th>
</tr>
</thead>
<tbody>
<tr>
<td>ssDelear</td>
<td>9999</td>
</tr>
<tr>
<td>ssTampered</td>
<td>9998</td>
</tr>
</tbody>
</table>

ServerTimeBias
Returns the difference, in hours, between local time and Greenwich Mean Time.

ServerType
Returns the type of server from the following list of valid ServerType enumerated values.

Table 5.6. ServerType Enumerated Values

<table>
<thead>
<tr>
<th>Enumerated value</th>
<th>Numeric equivalent</th>
</tr>
</thead>
<tbody>
<tr>
<td>sOS2</td>
<td>0x0CF1L</td>
</tr>
<tr>
<td>sNT</td>
<td>0x0CF2L</td>
</tr>
</tbody>
</table>

ServerVersion
Returns the version number of the server in hexadecimal format. Use ServerVersionAsString to view the server version in text format.

ServerVersionAsString
Returns the version number of the server as a text string. Use ServerVersion to view the server version in hexadecimal format.

TimeZoneInfo
Returns information about the time zone that the server is using for the current time. The time zone will be one of the following TimeZoneInfoType enumerated values.
Table 5.7. TimeZoneInfoType Enumerated Values

<table>
<thead>
<tr>
<th>Enumerated value</th>
<th>Numeric equivalent</th>
</tr>
</thead>
<tbody>
<tr>
<td>tzilnvalid</td>
<td>0</td>
</tr>
<tr>
<td>tziValid</td>
<td>1</td>
</tr>
<tr>
<td>tziStandardTime</td>
<td>2</td>
</tr>
<tr>
<td>tziDaylightSavingsTime</td>
<td>3</td>
</tr>
</tbody>
</table>

**ValidateBillingCodes**
A Boolean property, this indicates whether or not the billing codes will be checked against a list of valid codes when submitted. True or 1 indicates they will be validated; False or 0 indicates that they will not be validated.

**Methods**
None.
The FaxServer and ServerInfo Objects

Exercise

Add the following functionality to the sample application from the tutorial.

1. Create a ServerInfo object.
   • Interrogate the following ServerInfo object properties:
     – ValidateBillingCodes
     – IsEnterprise
   • Create text or list boxes to display the results in your application.
Section 6
Fax Objects and Attachments

Objectives

Upon completion of this section, you should be able to:

- Describe the two types of Fax objects.
- Utilize the attachment object for an inbound fax.
- Associate an attachment object with an outbound fax.
Fax Objects and Attachments

Fax Object Overview

There are two types of Fax objects. When a fax is received into the system, a Fax object is created that contains the sender and recipient information. This object will have an Attachment object that contains the name of the page image of the incoming fax in TIFF format.

When a user creates a fax to be sent to someone else, the new Fax object can reference other objects such as a signature file or cover sheet. In addition, there can be several types of Attachment objects (such as billing codes). When the user sends the fax, the Fax object will be converted to a page image in TIFF format, and the name of the image will be stored in an Attachment object.

This section describes the Fax object and the associated properties and methods. It also describes the objects that can be attached to a Fax object (Files, LibraryDocuments, Forms, and BillingCodes).

Attachment

An Attachment object defines the objects associated with a specific Fax. There are four types of Attachment objects, distinguished by the value of the AttachmentType property. When creating a new fax, an attachment can be either a form, billing code, file or library document. For incoming faxes, the AttachmentType is always File.

Properties

AttachmentType
Specifies the type of attachment from the following list of valid AttachmentType enumerated values. For each AttachmentType, you must also specify the property associated with that type.
Table 6.1 AttachmentType Enumerated Values

<table>
<thead>
<tr>
<th>Name</th>
<th>Numeric equivalent</th>
<th>Associated property</th>
<th>See also</th>
</tr>
</thead>
<tbody>
<tr>
<td>aFile</td>
<td>1</td>
<td>FileName</td>
<td>File object</td>
</tr>
<tr>
<td>aLibraryDocument</td>
<td>2</td>
<td>LibraryDocument</td>
<td>LibraryDocument object</td>
</tr>
<tr>
<td>aForm</td>
<td>3</td>
<td>Form</td>
<td>Form object</td>
</tr>
</tbody>
</table>

For incoming faxes, the system creates an AttachmentType of File for the image of the fax.

**BillingCode**
Specifies the code used for accounting purposes. This property is only used when the AttachmentType is aBillingCode.

**DeleteAfterSend**
This Boolean property indicates whether or not the attachment object should be deleted after it is copied to the fax server. There is no default. You must specify either True (deletes the object) or False.

**FileName**
Specifies the full path and name of the file that is to be attached to the Fax object. For incoming faxes, this property names the file that contains the image of the incoming fax. This property is only used when the AttachmentType is aFile.

**Form**
Specifies the name of the form that is to be overlaid onto the generated fax (such as company letterhead or a transmission form). This property is only used when AttachmentType is aForm.

**LibraryDocument**
Specifies the name of an existing document to be attached to the Fax object (such as company literature or employment applications). This property is only used when AttachmentType is aLibraryDocument.

**Methods**
None.
Fax Objects and Attachments

See Also
Attachments Collection, BillingCode, Form, LibraryDocument

Example

The following example stores the name of the first attachment to a variable called sTmp FileName.

```vba
Dim obFax As RCOMAPILib.Fax
Dim obAttachment As RCOMAPILib.Attachment
Dim sTmpFileName As String
Set obFax = MyFaxServer.User("Owner of Fax").Faxes(1)
Set obAttachment = obFax.Attachments(1)
sTmpFileName = obAttachment.FileName
'Do something with sTmpFileName
```

Attachments Collection

An Attachments collection contains one or more Attachment objects.

Properties

- **Count**
  Returns the total number of objects in the collection.

- **Create**
  Returns a new Attachment object.

- **Item**
  Returns the Attachment object with the specified Item number. Each object in the collection is numbered, starting with 1. For example, to check for a second attachment associated with a fax, enter:

  ```vba
  attachment.item (2)
  ```

Methods

- **Add**
  Adds an object to the collection.
Remove
Removes a specified type of attachment object from the collection but not from the server. To remove an object from the server, use the Delete method in the object you want to delete. You must specify one of the valid AttachmentType Enumerated Values.

RemoveAll
Removes all objects from the collection but not from the server. To remove an object from the server, use the Delete method in the object you want to delete.

See Also
Attachment, BillingCode, Form, LibraryDocument

Example
The following example checks to see if the fax has a library document attachment. If it does, this code adds the specified document to the fax before sending it.

```vbnet
If lstLibDocs.Text = "" Then
    oFax.Send
Else
    libdoc = lstLibDocs.Text
    Fax.Attachment.Add_
    MyFaxServer.LibraryDocuments(libdoc)
    oFax.Send
End If
```

BillingCode
A BillingCode object specifies categorization information that can be used later for functions such as accounting or tracking. For example, billing codes can be assigned to various clients, departments, or types of expense by your accounting department. These codes can be verified against a master billing code table within the RightFax system or used by external applications such as accounting.

Properties

BillInfo1
Specifies a string that represents a category in the system.

BillInfo2
Specifies a second string. A BillingCode object can contain either, both, or no BillInfo properties.
Fax Objects and Attachments

Description
Specifies a text description of the billing code.

Handle
The handle is used internally by the RightFax system to identify each unique object. If you specify a value without specifying a property name, the system will assign that value as the object Handle.

Methods

Delete
Removes the object from the server.

Save
Writes the object to the server.

See Also

BillingCodes Collection, Attachment

Example

The following example retrieves information on faxes related to a specific billing code and saves it to a database. This code also adds one extra page for the coversheet which is not counted in the TotalPages property of a sent fax.

Public Sub getPages()
Dim Fax As RFCOMAPILib.Fax
Dim nCount As Integer
For Each User in myFaxServer.Users
    For Each Fax in myFaxServer.Faxes(User.ID)
        bcCompare = Fax.BillingCode.BillInfo1
        If bcCompare = "BillingCode" Then
            nCount = nCount + (Fax.TotalPages + 1)
        End If
    Next
Next
End Sub
BillingCodes Collection

BillingCodes collection is a set of one or more BillingCode objects. This type of collection can get very large, depending on the complexity of the BillingCodes. Use the SearchOnKey, SearchString, and MaxRecords properties in conjunction with the Execute method to identify specific BillingCode objects within the collection.

Properties

**Count**
Returns the number of objects in the collection.

**Create**
Returns a new object.

**Item**
Returns the BillingCode object with the specified Item number. Each object in the collection is numbered, starting with 1.

**MaxRecords**
Indicates the maximum number of BillingCodes to return when searching with the Execute method.

**SearchOnKey**
Specifies the field to search for a BillingCode object within the collection. Valid key values are:

<table>
<thead>
<tr>
<th>Value</th>
<th>Field</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>BillInfo1</td>
</tr>
<tr>
<td>2</td>
<td>BillInfo2</td>
</tr>
<tr>
<td>3</td>
<td>Description</td>
</tr>
</tbody>
</table>

**SearchString**
Specifies the string for which to search in the specified SearchOnKey.
Fax Objects and Attachments

Methods

Add
Adds an object to the collection.

Execute
Begins a search for a particular BillingCode object, based on the values in the SearchOnKey and SearchString properties, and returns the objects that meet the specified criteria. For example:

Private Sub Form_Load()
    Set BillCodes = frmMain.g_FaxAPI.BillingCodes
    listBillCodes.ColumnHeader.Add , , “BI1”, 2000
    listBillCodes.ColumnHeader.Add , , “Descript”, 2000
    BillCodes.SearchString = “”
    BillCodes.SearchOnKey = 1
    BillCodes.MaxRecords = 999999
    BillCodes.Execute
    LoadBillCodes
End Sub

Remove
Removes a specified object from the collection but not from the server. To remove an object from the server, use the Delete method in the object you want to delete.

RemoveAll
Removes all objects from the collection but not from the server. To remove an object from the server, use the Delete method in the object you want to delete.

See Also
BillingCode, Attachment
Example

The following code gets the billing code collection and displays the code and description for each BillingCode1.

Private Sub Form_Load()
    Set BillCodes = frmMain.g_FaxAPI.BillingCodes
    listBillCodes.ColumnHeaders.Add , , "BI1", 2000
    listBillCodes.ColumnHeaders.Add , , "Description", 2000
    listBillCodes.ColumnHeaders.Add , , "Handle", 2000
    BillCodes.SearchString = ""
    BillCodes.SearchOnKey = 1
    BillCodes.MaxRecords = 999999
    BillCodes.Execute
    LoadBillCodes
End Sub

Private Sub LoadBillCodes()
    Dim BillCode As RFCOMAPILib.BillingCode
    Dim lstItem As ListItem
    listBillCodes.ListItems.Clear
    For Each BillCode In BillCodes
        Set lstItem = listBillCodes.ListItems.Add
        lstItem.Text = BillCode.BillInfo1
        lstItem.SubItems(1) = BillCode.Description
        lstItem.SubItems(2) = BillCode.Handle
    Next
End Sub

Private Sub listBillCodes_DblClick()
    Dim BillCode As RFCOMAPILib.BillingCode
    For Each BillCode In BillCodes
        If (BillCode.Handle = listBillCodes.SelectedItem.SubItems(2)) Then
            Set frmMain.g_CurBillCode = BillCode
            Unload Me
            Exit Sub
        End If
    Next
End Sub

Private Sub Form_Resize()
    listBillCodes.Width = frmBillCodes.Width - 120
    listBillCodes.Height = frmBillCodes.Height - 400
End Sub
Fax Objects and Attachments

CoverSheet

A CoverSheet object can be either a Microsoft Word document or a PCL file. When the fax is sent, the sender and recipient information as well as any cover sheet notes will be merged with the cover sheet template and converted to an image file for transmission. In the Fax object, the HasCoversheet property must be set to True or 1, and the file name must be specified with the FCSFilename property in order for the cover sheet to be sent. The cover sheet notes, if any, are specified with the CoverSheetNotes property for each CoverSheets Collection object.

Properties

ID
Specifies an ID you can assign to identify each unique object. If you specify a value without specifying a property name, the system will assign that value as the object ID.

Methods

None.

CoverSheets Collection

A CoverSheets collection is a set of one or more CoverSheet objects.

Properties

Count
Returns the number of objects in the collection.

Create
Returns a new object.

Item
Returns the Fax object with the specified item number. Each object in the collection is numbered, starting with 1.

Methods

Refresh
Re-loads the list of fax objects from the fax server.
Fax Objects and Attachments

Remove
Removes a specified object from the collection but not from the server. To remove an object from the server, use the Delete method in the object you want to delete.

RemoveAll
Removes all objects from the collection but not from the server. To remove an object from the server, use the Delete method in the object you want to delete.

Example
To specify a cover page called FaxLetterhead.pcl, use the following code.

```plaintext
myNewFax.FCSFilename = “FaxLetterhead.pcl”
myNewFax.HasCoversheet = True
```

Fax
A Fax object contains all the information about the components of a fax. There can be many other objects referenced by the properties in a single fax object. The default property name is Handle. If you specify a value without specifying a property name, the system will assign that value as the object Handle.

Properties

Attachments
Returns the Attachments collection associated with the current fax object. See Attachment object or Attachments Collection for more information.

AutoForwardCount
Returns the number of times that the fax has been forwarded using the AutoForward feature. This number is tracked for each fax in order to prevent infinite forwarding loops. By default the maximum number of allowed AutoForward events is 5.

BFTFilename
Returns the file name of the binary file transfer information file if one exists. Binary file transfers have been replaced by the fax image attachment, but this property remains available for backwards compatibility.

BFTFileSize
Returns the size of the binary file transfer information file if one exists. Binary file transfers have been replaced by the fax image attachment, but this property remains available for backwards compatibility.
Fax Objects and Attachments

BillingCode
Returns the BillingCode object associated with the fax, if any. See the BillingCode object for more information.

ChannelToSendOn
Specifies the channel on which to send the current fax.

CoverSheetNotes
Specifies a specific note line for the cover sheet. The note can have up to 24 lines, identified by Index number starting with 0. See also the CoverSheet object.

CoverSheetSizeInBytes
Returns the size of the fax cover sheet in bytes.

DelayFaxSendDateTime
Specifies the date and time that the fax should be sent in Visual Basic variant format.

FaxDisposition
Returns an integer that indicates the disposition of the fax. This property has been replaced by information in the FaxStatus property, but remains available for backwards compatibility.

FaxErrorCode
Returns the FaxErrorCodeType. A zero indicates that there was no error. The FaxErrorCodeType will be one of the following enumerated values.

<table>
<thead>
<tr>
<th>Name</th>
<th>Numeric equivalent</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>fecNone</td>
<td>0</td>
<td>No error</td>
</tr>
<tr>
<td>fecBusy</td>
<td>1</td>
<td>Fax number was busy</td>
</tr>
<tr>
<td>fecTransmissionError</td>
<td>2</td>
<td>Fax was not sent/received</td>
</tr>
<tr>
<td>fecPoorQuality</td>
<td>3</td>
<td>The fax board returned a message indicating that although the fax was received, it contained errors</td>
</tr>
<tr>
<td>fecNoAnswer</td>
<td>4</td>
<td>The destination number did not answer the phone</td>
</tr>
<tr>
<td>fecBadFCS</td>
<td>5</td>
<td>Bad File Cover Sheet</td>
</tr>
<tr>
<td>fecBadConvert</td>
<td>6</td>
<td>The fax could not be converted to an image file</td>
</tr>
<tr>
<td>fecMakeFCS</td>
<td>7</td>
<td>Error while creating Fax Cover Sheet</td>
</tr>
</tbody>
</table>

Table 6.3 FaxErrorCodeType Enumerated Values
Table 6.3 FaxErrorCodeType Enumerated Values (Continued)

<table>
<thead>
<tr>
<th>Name</th>
<th>Numeric equivalent</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>fecCantSchedule</td>
<td>8</td>
<td>Fax was not sent because fax transmission schedule full</td>
</tr>
<tr>
<td>fecUnknown</td>
<td>9</td>
<td>Unknown error report by fax board</td>
</tr>
<tr>
<td>fecHuman</td>
<td>10</td>
<td>A human answered the phone</td>
</tr>
<tr>
<td>fecGroup2</td>
<td>11</td>
<td>Reached a Group2 machine</td>
</tr>
<tr>
<td>fecLocalInUse</td>
<td>12</td>
<td>The destination fax number is unable to receive</td>
</tr>
<tr>
<td>fecLineProblem</td>
<td>13</td>
<td>Communications line between the two fax numbers not working correctly</td>
</tr>
<tr>
<td>fecBadPaper</td>
<td>14</td>
<td>Invalid FormType specified</td>
</tr>
<tr>
<td>fecBadSignature</td>
<td>15</td>
<td>Invalid signature file specified</td>
</tr>
<tr>
<td>fecNoSignature Authorization</td>
<td>16</td>
<td>The fax required but did not receive signature authorization</td>
</tr>
<tr>
<td>fecDiscarded</td>
<td>18</td>
<td>Canceled by user</td>
</tr>
<tr>
<td>fecBadPhone</td>
<td>19</td>
<td>Invalid characters in fax number</td>
</tr>
<tr>
<td>fecInvalidCode</td>
<td>21</td>
<td>Invalid BillingCode</td>
</tr>
<tr>
<td>fecBadCode</td>
<td>22</td>
<td>Error in an embedded code</td>
</tr>
<tr>
<td>fecBadOCR</td>
<td>23</td>
<td>OCR operation failed</td>
</tr>
<tr>
<td>fecBadPrint</td>
<td>24</td>
<td>Print operation failed</td>
</tr>
<tr>
<td>fecNoLibraryDocument Authorization</td>
<td>25</td>
<td>Unauthorized user attempting to create a new library document</td>
</tr>
<tr>
<td>fecViewStar1</td>
<td>26</td>
<td>ViewStar imaging system import error</td>
</tr>
<tr>
<td>fecDisapproved</td>
<td>27</td>
<td>The fax was disapproved for transmission</td>
</tr>
<tr>
<td>fecEmailDeliveryError</td>
<td>28</td>
<td>The fax document could not be sent to the specified e-mail address</td>
</tr>
</tbody>
</table>

**FaxFilename**

Returns the name of the image file associated with an incoming fax. The image file is stored in the image folder on the fax server. Note that this property returns the file name, not the image object.

**FaxID**

Returns the ID assigned to the fax by the fax hardware when the fax is schedule to be sent.
Fax Objects and Attachments

FaxPagesInFront
Returns the number of fax pages in the queue preceding the current fax.

FaxRecordDateTime
Returns the date and time that the fax was created in Visual Basic variant format.

FaxStatus
Returns the status of the fax as a FaxStatusType. This will be one of the valid FaxStatusType enumerated values.

<table>
<thead>
<tr>
<th>Enumerated value</th>
<th>Numeric equivalent</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>fsUnborn</td>
<td>0</td>
<td>Not yet created</td>
</tr>
<tr>
<td>fsNeedsFCS</td>
<td>1</td>
<td>Needs fax cover sheet</td>
</tr>
<tr>
<td>fsNeedsConversion</td>
<td>2</td>
<td>Needs conversion to an image file</td>
</tr>
<tr>
<td>fsNeedsToSend</td>
<td>3</td>
<td>None</td>
</tr>
<tr>
<td>fsInConversion</td>
<td>4</td>
<td>In the process of being converted to an image file</td>
</tr>
<tr>
<td>fsInSend</td>
<td>5</td>
<td>usPagesInFront indicates how many pages sent so far</td>
</tr>
<tr>
<td>fsDoneOK</td>
<td>6</td>
<td>Successfully sent or received</td>
</tr>
<tr>
<td>fsManualFCS</td>
<td>7</td>
<td>Uses a manual fax cover sheet</td>
</tr>
<tr>
<td>fsInSchedule</td>
<td>8</td>
<td>Awaiting scheduled send time</td>
</tr>
<tr>
<td>fsDoneError</td>
<td>9</td>
<td>Too many errors, will not be retried</td>
</tr>
<tr>
<td>fsDuplicate</td>
<td>10</td>
<td>None</td>
</tr>
<tr>
<td>fsError</td>
<td>11</td>
<td>None</td>
</tr>
<tr>
<td>fsNeedsAttention</td>
<td>12</td>
<td>None</td>
</tr>
<tr>
<td>fsNeedsAttachment</td>
<td>13</td>
<td>None</td>
</tr>
<tr>
<td>fsHeldForPreview</td>
<td>14</td>
<td>None</td>
</tr>
<tr>
<td>fsInOCR</td>
<td>15</td>
<td>None</td>
</tr>
<tr>
<td>fsInPrint</td>
<td>16</td>
<td>None</td>
</tr>
<tr>
<td>fsQueuedForPrinting</td>
<td>17</td>
<td>None</td>
</tr>
<tr>
<td>fsQueuedForOCR</td>
<td>18</td>
<td>None</td>
</tr>
<tr>
<td>fsInValidation</td>
<td>19</td>
<td>None</td>
</tr>
<tr>
<td>fsInApproval</td>
<td>20</td>
<td>None</td>
</tr>
</tbody>
</table>
FaxTerminationStatus
Returns an integer indicating the termination status of the fax. The information available through this property has been moved to the FaxHistory object but this property remains available for backwards compatibility. See the Histories collection or the FaxStatus property.

FCSFilename
Specifies the name of the fax cover sheet image in the server’s image folder. See also the OriginalFCSFilename property.

Folder
Specifies the Folder object to which this fax belongs.

ForwardToNewFaxNumber
Returns a new fax that has all of the same properties as the previous fax except the FaxID.

FromFaxNumber
Specifies the sender’s personal fax number.

FromGeneralFaxNumber
Specifies the sender’s general fax number.

FromGeneralVoiceNumber
Specifies the sender’s general voice telephone number.

FromName
Specifies the sender’s name.

FromVoiceNumber
Specifies the sender’s personal voice telephone number.

GenerateCompletionEvent
This Boolean property specifies whether or not to generate a completion event once the fax has been sent. Set this property to True or 1 to generate the CompletionEvent. Set it to False or 0 if you don’t want to generate a CompletionEvent.

Handle
The handle is used internally by the RightFax system to identify each unique object. If you specify a value without specifying a property name, the system will assign that value as the object Handle.
Fax Objects and Attachments

**HasBFT**
A Boolean property, this indicates whether or not the fax will use binary file transfer. The property returns True or 1 if the fax has BFT; it returns False or 0 if it does not. Binary file transfers have been replaced by the fax image attachment, but this property remains available for backwards compatibility.

**HasCoversheet**
A Boolean property, this indicates whether or not the fax has a cover sheet. It is only valid for outgoing faxes. True or 1 indicates there is a cover sheet; False or 0 indicates there is no cover sheet. The cover sheet is stored as a file on the server and can be specified using the FCSFilename property. Each user can specify a default cover page setting with the IsCoverPageDefaultedOn property, or a group may be required to use a cover sheet based on the MustHaveCoversheet property.

**HasPDF**
A Boolean property, this indicates whether or not the fax has a rich PDF document associated with it.

**HideFromWeb**
A Boolean property, this hides the fax from the certified delivery Web interface when the recipient deletes it. (The original fax cannot be deleted from the fax database by the recipient, so it is hidden instead.)

**Histories**
Returns the FaxHistories collection associated with the fax. The collection will contain one or more history objects.

**ImageSizeInBytes**
Returns the size of the fax body in bytes.

**InputFilename**
Returns the name of the file that appears in the outgoing folder. This property has been replaced by functionality in the Attachment object. See also Attachments Collection.

**IsApproved**
A Boolean property, this indicates whether or not the fax has been approved. True or 1 indicates it has been approved; False or 0 indicates it has not been approved.

**IsBillingCodesVerified**
A Boolean property, this indicates whether or not the BillingCode attachment has been checked against the list of valid billing codes. True or 1 indicates the BillingCodes have been checked and are valid. False or 0 indicates that they are not valid billing codes.
Fax Objects and Attachments

IsBroadcastFax
This Boolean property specifies whether or not the fax is a broadcast fax. True or 1 indicates that the fax is a broadcast fax, and the status will not be updated until the entire job is complete (thus saving machine overhead).

IsCallbackRequested
This Boolean property indicates whether or not the sender requests a call from the recipient. True or 1 indicates that a callback is requested; False or 0 indicates that no such request has been made. This property only applies to sent faxes. The default setting for callback request is set with the IsCallbackRequested property in the User object.

IsCertifyDelivered
A Boolean property, this indicates whether or not the fax was sent via certified delivery.

IsCoversheetConversionNeeded
This Boolean property specifies whether or not the cover sheet for the fax needs to be converted to a TIFF format image file. True or 1 indicates that conversion is required; False or 0 indicates that it is not required.

IsCoversheetFine
This Boolean property specifies whether or not the cover sheet is in Fine mode. True or 1 indicates that the cover sheet is in Fine mode (200×200 dots per inch resolution); False or 0 indicates that it is in normal mode (100×100 dots per inch resolution). See also the IsFineMode property.

IsDeleted
This Boolean property specifies whether or not the fax has been deleted. True or 1 indicates that the fax has been deleted from the server. False or 0 indicates that it has not been deleted.

IsDeletedAfterAnySend
This Boolean property specifies whether or not the fax will be deleted from the server after the Send event, whether the fax is successfully sent or not. Set this to True or 1 to indicate that the fax will be deleted. Set it to False or 0 to indicate that it will not be deleted. See also the IsDeletedAfterSuccessfulSend property.

IsDeletedAfterSuccessfulSend
This Boolean property specifies whether or not the fax will be deleted from the server after a successful Send event. Set this to True or 1 to indicate that the fax will be deleted. Set it to False or 0 to indicate that the fax will not be deleted. See also the IsDeletedAfterAnySend property.
Fax Objects and Attachments

**IsFaxAutoForwarded**
This Boolean property specifies whether or not the fax was automatically forwarded. The AutoForward feature is set and type of forwarding defined in the User object. True or 1 indicates that the fax was automatically forwarded. False or 0 indicates that the fax was not forwarded. See also the IsForwarded property.

**IsFaxBodyAutomaticallyPrinted**
This Boolean property specifies whether or not the fax was automatically printed. The Autoprint feature is set and the printer defined in the User object. True or 1 indicates that this fax was automatically printed. False or 0 indicates that this fax was not forwarded.

**IsFCSCompleted**
This Boolean property specifies whether or not the fax cover sheet has been completed. True or 1 indicates that the cover sheet is complete. False or 0 indicates that the cover sheet is not complete.

**IsFineMode**
This Boolean property specifies whether or not the body of the fax is in Fine mode. True or 1 indicates that the fax is in Fine mode (200×200 dots per inch resolution); False or 0 indicates that it is in normal mode (100×100 dots per inch resolution). See also the IsCoversheetFine property.

**IsForwarded**
This Boolean property specifies whether or not the fax was manually forwarded. True or 1 indicates that the fax was forwarded. False or 0 indicates that the fax was not forwarded. Use the ForwardToUsers method to specify manual forwarding.

**IsGatewayGenerated**
This Boolean property specifies whether or not the fax was generated by the RightFax e-mail gateway. True or 1 indicates that the fax was generated by an e-mail gateway. False or 0 indicates that it was not.

**IsGenericFlag1On**
This Boolean property specifies whether or not the user customizable API flag Generic1 is on.

**IsGenericFlag2On**
This Boolean property specifies whether or not the user customizable API flag Generic2 is on.

**IsHeld**
This Boolean property specifies whether or not the fax is being held until the user previews it. True or 1 indicates that the fax is awaiting preview. False or 0 indicates that it is not.
**Fax Objects and Attachments**

**IsInDelaySend**
This Boolean property specifies whether or not the fax is being held until the date and time specified by the DelayFaxSendDateTime property. True or 1 indicates that the fax is awaiting the specified time. False or 0 indicates that it is not.

**IsInitialized**
This Boolean property specifies whether or not the fax was initialized. True or 1 indicates that the fax was initialized. False or 0 indicates that the fax was not initialized.

**IsLCRTimeDelayed**
This Boolean property specifies whether or not the fax was delayed because of a least-cost routing rule. True or 1 indicates that the fax was delayed. False or 0 indicates that the fax was not delayed.

**IsNeedingApproval**
This Boolean property specifies whether or not the fax is being held until it is approved. True or 1 indicates that the fax is awaiting approval. False or 0 indicates that it is not. Use the Approve method to send the fax.

**IsPrinted**
This Boolean property specifies whether or not the fax was successfully printed. True or 1 indicates that the fax was printed. False or 0 indicates that the fax was not printed. See also the PrintFax method.

**IsReceived**
This Boolean property specifies whether the fax is an incoming or outgoing fax. True or 1 indicates that the fax was received (incoming). False or 0 indicates that the fax was sent (outgoing).

**IsViewed**
This Boolean property specifies whether or not the fax was viewed. True or 1 indicates that the fax was viewed. False or 0 indicates that the fax was not viewed.

**LastHistoryChangeDateTime**
This returns the date and time that the most recent history element was added to this fax. It is updated every time an element is added to any of the fax history objects. See also the Histories collection.

**MaximumRetries**
This property specifies the maximum number of times the system should attempt to deliver the fax if the phone number is not responding. Use the TryInterval property to specify the length of time between retries.
Fax Objects and Attachments

**NeedsPDFConversion**
A Boolean property, this indicates whether or not the fax requires PDF conversion. When PDF conversion is complete, this property is automatically set to False or 0.

**OriginalBFTFilename**
Returns the name of the binary file transfer file before it was renamed by the fax server. Binary file transfers have been replaced by the fax image attachment, but this property remains available for backwards compatibility.

**OriginalFCSFilename**
Returns the name of the original file that was converted to use as the fax cover sheet. This is the name as it appears in the server folder. The converted file name is accessible through the FCSFilename property.

**Owner**
Returns the User object for the owner of the fax.

**OwnerID**
Returns the UserID for the owner of the fax.

**Priority**
Sets the priority for the fax using one of the valid FaxPriorityType enumerated values from the following table. Note that in order for the user to send a fax with High priority, the IsAbleToUseHighPriority property must be set to True or 1 and the HighestAvailablePriority property must be set to fpHigh in the User object.

<table>
<thead>
<tr>
<th>Name</th>
<th>Numeric equivalent</th>
</tr>
</thead>
<tbody>
<tr>
<td>fpNormal</td>
<td>0</td>
</tr>
<tr>
<td>fpLow</td>
<td>1</td>
</tr>
<tr>
<td>fpHigh</td>
<td>2</td>
</tr>
</tbody>
</table>

**Recipients**
Specifies multiple users to whom the fax will be sent. Define the users in a PhoneItemGroup, then specify that group as recipients. See also the PhoneItemGroup object.

**RemoteID**
Returns the ID of the sender’s fax machine. This applies only to received faxes.
**SecureCSID**
Returns the secure caller subscriber identification of the sender’s fax machine, if any. This applies only to received faxes.

**SendTime**
Returns the total length of transmission time it took to send or receive the fax. This is available for all received faxes. It is only available for sent faxes after successful completion.

**StatusDescription**
Returns the current status of the fax in text format. For a numeric code indicating the status, see the FaxStatus property.

**ToCityState**
Specifies the city and state of the recipient. See also the DefaultToCityState property in the User object.

**ToCompany**
Specifies the recipient’s company name. See also the DefaultToCompany property in the User object.

**ToFaxNumber**
Specifies the telephone number of the recipient’s fax machine. See also the DefaultToFaxNumber property in the User object.

**ToName**
Specifies the recipient’s name. See also the DefaultToName property in the User object.

**TotalLinkCount**
Returns the total number of links associated with this fax. Links are generated when a RightFax user sends or forwards a fax to another RightFax user. Rather than duplicate the image file of the fax for each user, the system creates a link to a single instance of the image.

**TotalPages**
Returns the total number of pages in the fax. The fax must be converted for this to be valid.

**ToVoiceNumber**
Specifies the telephone number of the recipient. See also the DefaultToVoiceNumber property in the User object.

**TryInterval**
Specifies how long to wait, in seconds, between attempts to send the fax. The MaximumRetries property must be set to greater than 0 for this to have any effect.
Fax Objects and Attachments

**TypeOfPrintJob**
Specifies the type of data in the print job. Use one of the following PrintJobType enumerated values.

<table>
<thead>
<tr>
<th>Name</th>
<th>Numeric equivalent</th>
</tr>
</thead>
<tbody>
<tr>
<td>pjPCL</td>
<td>1</td>
</tr>
<tr>
<td>pjPostScript</td>
<td>2</td>
</tr>
<tr>
<td>pjPCL2</td>
<td>3</td>
</tr>
<tr>
<td>pjPostScript2</td>
<td>4</td>
</tr>
<tr>
<td>pjCVL</td>
<td>5</td>
</tr>
</tbody>
</table>

**UniqueID**
Specifies the unique ID for this specific print job. This can be defined by either the system or the user.

**UserComments**
Comment text added to the fax by the fax sender.

**UseSmartResume**
This Boolean property indicates whether or not the fax will use the Smart Resume feature of the RightFax system. True or 1 indicates the fax will use Smart Resume. False or 0 indicates it will not.

**Methods**

**Approve**
This method is only available if the IsNeedingApproval property is set to True or 1. The Approve method sets the IsNeedingApproval property to False or 0 and the IsApproved property to True or 1, then submits the fax to the queue to be sent.

**Delete**
Removes the fax from the server. To remove a fax from the Faxes collection but not from the server, use the Remove or RemoveAll methods at the collection level.
**Disapprove**
This method is only available if the IsNeedingApproval property is set to True or 1. The Disapprove method sets the IsNeedingApproval property to False or 0 and the IsApproved property to False or 0, indicating that the fax was not approved and will not be sent.

**ForwardToUsers**
Sends the fax to multiple users within the RightFax system. For example:

```vba
Dim myUsers As Users
Set myUsers = objFaxServer.Users
myUsers.RemoveAll
myUsers.Add myFaxServer.User("USER1")
myUsers.Add myFaxServer.User("USER2")
objFax.ForwardToUsers myUsers, "Check out this fax"
```

**Kick**
Re-sends a fax that has already been sent. See also the Send method for new faxes.

**Mark**
Changes the state of a fax to a specified FaxMarkType value.

---

**Table 6.7 FaxMarkType Enumerated Values**

<table>
<thead>
<tr>
<th>Name</th>
<th>Numeric equivalent</th>
</tr>
</thead>
<tbody>
<tr>
<td>fmViewed</td>
<td>0</td>
</tr>
<tr>
<td>fmUnViewed</td>
<td>1</td>
</tr>
<tr>
<td>fmPrinted</td>
<td>3</td>
</tr>
<tr>
<td>fmNotPrinted</td>
<td>4</td>
</tr>
<tr>
<td>fmReleased</td>
<td>5</td>
</tr>
<tr>
<td>fmHeld</td>
<td>6</td>
</tr>
<tr>
<td>fmBodyPrinted</td>
<td>7</td>
</tr>
<tr>
<td>fmApproved</td>
<td>8</td>
</tr>
<tr>
<td>fmDisapproved</td>
<td>9</td>
</tr>
<tr>
<td>fmGeneric1</td>
<td>10</td>
</tr>
<tr>
<td>fmNotGeneric1</td>
<td>11</td>
</tr>
<tr>
<td>fmGeneric2</td>
<td>12</td>
</tr>
<tr>
<td>fmNotGeneric2</td>
<td>13</td>
</tr>
</tbody>
</table>
For example, to mark a record as approved, enter the following line of code.

```csharp
objFax.Mark fmApproved
```

This method has an Atomic parameter, which allows you to block other transactions while this one is taking place. As soon as the method is complete, the object will be unblocked. Set the Atomic parameter to True or 1 to block; set it to False or 0 to allow access. The default is False or 0.

**OCR**

Converts the current fax from an image to text using optical character recognition algorithms. Inputs are the start page, the end page, the OCRFormatType, OCRLayoutType, and a file extension for the resulting file (such as .txt or .doc). If you don’t specify OCRFormatType or OCRLayoutType with this method, then the system will use the format specified in the User object. See also AutoOCRLayout.

<table>
<thead>
<tr>
<th>Name</th>
<th>Numeric equivalent</th>
</tr>
</thead>
<tbody>
<tr>
<td>ofSmartASCII</td>
<td>0</td>
</tr>
<tr>
<td>ofASCII</td>
<td>1</td>
</tr>
<tr>
<td>ofRTF</td>
<td>3</td>
</tr>
<tr>
<td>ofUseUserType</td>
<td>0xFFFF (See also AutoOCRFormat)</td>
</tr>
</tbody>
</table>

**Table 6.8 OCRFormatType Enumerated Values**

<table>
<thead>
<tr>
<th>Name</th>
<th>Numeric equivalent</th>
</tr>
</thead>
<tbody>
<tr>
<td>olWYSIWYG</td>
<td>0</td>
</tr>
<tr>
<td>olLeftJustified</td>
<td>1</td>
</tr>
<tr>
<td>olUseUserType</td>
<td>0xFFFF</td>
</tr>
</tbody>
</table>

**Table 6.9 OCRLayoutType Enumerated Values**
PrintFax
This method sends the fax to the print queue to await printing. Inputs are the start page (default is -1), end page, and a Boolean operator (True or False) that indicates whether or not to include the cover page.

RouteToUser
Sends a fax to a user within the RightFax system. To send to multiple users, use the ForwardToUsers method.

Save
Saves the fax object. The Save method also allows you to automatically re-send a fax by using the Kick parameter. Set the parameter to True or 1 to send the fax after it is saved; set the parameter to False or 0 if you want to save without sending. See also the Kick method, which resends the fax as a separate method.

Send
Converts the fax to an image file and then sends it to the specified recipients. To resend a fax that has already been sent, use the Kick method.

StoreAsFormUpdate
Stores the current fax as a form that can be used later by other fax objects. This method updates an existing form, which you must specify. To store the form to a new name and create a new form, see StoreAsNewForm. This method also has a parameter which specifies whether you want the form stored with the TTI information (the top half-inch of the fax) or not. Set the RemoveTTILine parameter to True or 1 to remove the TTI line. The default for the RemoveTTILine parameter is False or 0 (leave the form as is).

StoreAsLibraryDocumentUpdate
Stores the current fax as a library document that can be used later by other fax objects. This method updates an existing library document, which you must specify. See StoreAsNewLibraryDocument to create a new library document.

StoreAsNewForm
Stores the current fax as a new form that can be used later by other fax objects. You must specify a file name for the form. This method also has a parameter that specifies whether you want the form stored with the TTI information (the top half-inch of the fax) or not. Set the RemoveTTILine parameter to True or 1 to remove the TTI line. The default for the RemoveTTILine parameter is False or 0 (leave the form as is).
Fax Objects and Attachments

**StoreAsNewLibraryDocument**
Stores the current fax as a new library document that can be used later by other fax objects. You must specify a file name for the library document. For example, to create a document named PRICES with the description “Price Sheet,” enter the following line.

```
objFax.StoreAsNewLibraryDocument "PRICES", "Price Sheet"
```

**Example**

The following example sends a fax with an attached library document.

```
Set oFax = MyFaxServer.CreateObject(coFax)
oFax.Owner = oUser
oFax.ToName = tofaxname
oFax.ToFaxNumber = tofaxnum

If 1stLibDocs.Text = "" Then
  oFax.Send
Else
  libdoc=1stLibDocs.Text
  oFax.Attachment.Add_
  MyFaxServer.LibraryDocuments(libdoc)
  oFax.Send
EndIf
```
Exercise 1

In this exercise you will add the ability to send a fax in your sample application.

1. Create a Fax object.
   - From a text box assign information to the ToName, ToFaxNumber and Owner properties.
   - Create a button which will send the fax to the information supplied in the text boxes.
Fax Objects and Attachments

Faxes Collection

A Faxes collection is a set of one or more Fax objects.

Properties

Count
Returns the number of objects in the collection.

Create
Returns a new object.

Item
Returns the Fax object with the specified item number. Each object in the collection is numbered, starting with 1.
Methods

Add
Adds an object to the collection.

Refresh
Re-loads the list of fax objects from the fax server.

Remove
Removes a specified object from the collection but not from the server. To remove an object from the server, use the Delete method in the object you want to delete.

RemoveAll
Removes all objects from the collection but not from the server. To remove an object from the server, use the Delete method in the object you want to delete.

See Also
Fax, Attachment

Example

The following example steps through a collection of faxes for each user and displays the fax image.

```
Dim myFax As RCOMAPILib.Fax
For Each myFax in FaxApi.Faxes("{myUser}"
DisplayImage myFax.Attachments(1)
```

Form

A form is an object that overlays a fax, such as corporate letterhead. Within the RightFax system, the form is considered an attachment to a CoverSheets Collection object. To the recipient, the form appears to be part of the fax image.

Properties

Append
Specifies whether the form should be appended to the end of the fax pages (such as a disclaimer page for proposals) or should be used as an overlay on top of the fax page. True or 1 indicates the form should be added to the end of the fax. False or 0 indicates the form is an overlay. The default is False or 0.
Fax Objects and Attachments

**Handle**
Returns the code used internally by the RightFax system to identify each unique object.

**ID**
Specifies an ID you can assign to identify each unique object. If you specify a value without specifying a property name, the system will assign that value as the object ID.

**Name**
Specifies a descriptive name for the form.

**NextFormNum**
Specifies the number of the next form that will be attached to the current fax object. For example, if one form has the corporate logo and contact information for use on the first page of a document, the next form may have only the logo for use on the second and subsequent pages.

**Num**
Specifies the ordinal number of the form object within the Forms collection. For example, the first form in the collection is number 1, the second form added to the collection is number 2, etc.

**PageCount**
Returns the total number of pages in the form object.

**PageToStartOn**
Specifies the page of the fax on which to begin displaying the form. For example, if the form is designed for the second and subsequent pages of a letter, you might specify PageToStartOn(2).

**SecurityID**
Specifies the user or group authorized to access the form.

**ServerFileName**
Returns the name under which the form’s image files are stored on the server.

**Methods**

**Delete**
Removes the object from the server.

**Save**
Writes the object to the server.
See Also

Attachment, Forms Collection

Forms Collection

A Forms collection is a set of one or more Form objects.

Properties

Count
Returns the number of objects in the collection.

Create
Returns a new Form object. Note that the user must be authorized to create new forms by having the IsAbleToCreateNewForms property set to True or 1 in the user object.

Item
Returns the form object with the specified item number. Each object in the collection is numbered, starting with 1.

Methods

Add
Adds an object to the collection.

Refresh
Re-loads the list of form objects from the fax server.

Remove
Removes a specified object from the collection but not from the server. To remove an object from the server, use the Delete method in the object you want to delete.

RemoveAll
Removes all objects from the collection but not from the server. To remove an object from the server, use the Delete method in the object you want to delete.

See Also

Form
LibraryDocument

A library document is a pre-defined image object that can be attached to a fax before sending. For example, you can have a collection of sales literature available for faxing. Each document can be attached to a fax as a library document. Note that the AttachmentType must be set to aLibraryDocument in the Attachment object.

Properties

Description
Returns a brief description of the library document.

EmbargoDate
Specifies the earliest date a document can be used. For example, a price list document can be embargoed until the first of the following month. After that date, the library document can be sent. For an embargo date to be effective, the IsEmbargoed property must be set to True or 1. To specify the last date a document can be sent, see ExpireDate.

ExpireDate
Specifies the last date a document can be used. For example, sale prices may only be valid for a short time. After the specified date, the library document containing the sale prices would no longer be available. For an expiration date to be effective, the WillExpire property must be set to True or 1. To specify the first date a document can be sent, see EmbargoDate.

FODPassword
Specifies the password that must be entered before the form can be sent using the RightFax Docs-on-Demand module.

Handle
Returns the code used internally by the RightFax system to identify each unique object.

ID
Specifies an ID you can assign to identify each unique object. If you specify a value without specifying a property name, the system will assign that value as the object ID.

IsALibraryDocument
This property has been replaced by new functionality, but remains available for backward compatibility. All LibraryDocument objects have a value of True or 1 in this property.
**IsAutomaticallyGeneratedCatalog**
This Boolean property indicates whether or not the library document is a catalog or listing of other library documents. This type of catalog is part of the RightFax Docs-on-Demand module. True or 1 indicates the library document is a catalog; False or 0 indicates it is not.

**IsEmbargoed**
This Boolean property indicates whether or not the library document has a start date. See also the EmbargoDate property.

**IsIncludedInFODCatalog**
This Boolean property indicates whether or not the library document is included in the catalog created by the RightFax Docs-on-Demand module. True or 1 indicates the document is included; False or 0 indicates that it is not. The catalog created by the Docs-on-Demand module is also a library object, indicated by the property IsAutomaticallyGeneratedCatalog being set to True.

**IsInFineMode**
This Boolean property indicates whether or not the resolution of the library document is Fine. True or 1 indicates that the fax resolution is Fine (200×200 dots per inch); False or 0 indicates that it is normal (100×100 dots per inch). See also the IsFineMode property for the body of the fax or IsCoversheetFine for the cover sheet mode.

**IsInNormalMode**
This Boolean property indicates whether or not the resolution of the library document is fine. True or 1 indicates that the fax is normal (100×100 dots per inch); False or 0 indicates that it is fine (200×200 dots per inch). See also the IsFineMode property for the body of the fax or IsCoversheetFine for the cover sheet mode.

**IsPublishedForFOD**
This Boolean property indicates whether or not the library document is available for use with the RightFax Docs-on-Demand module. True or 1 indicates that the library document is available; False or 0 indicates that it is not. If True, see also the TimesUsedOnFOD property.

**IsPublishedForLAN**
This Boolean property indicates whether or not the library document is available to users of the local area network. True or 1 indicates that the library document is available; False or 0 indicates that it is not. If True, see also the TimesUsedOnLAN property.
Fax Objects and Attachments

**IsPublishedForWeb**
This Boolean property indicates whether or not the library document is available for use on the Web through the RightFax WebTools. True or 1 indicates that the library document is available; False or 0 indicates that it is not. If True, see also the TimesUsedOnWeb property.

**LastUsedDate**
Returns the most recent date that this library document was used. The last used date can be re-set using the AdjustUsage method.

**PageCount**
Returns the total number of pages in the library document.

**ServerFileName**
Returns the name of the library document image file as it exists on the server. Note that this property returns only the file name, not the document itself.

**TimesUsedOnFOD**
Returns the number of times the library document has been used by the RightFax Docs-on-Demand module. See the IsPublishedForFOD property to determine whether or not the library document is available for use with that module.

**TimesUsedOnLAN**
Returns the number of times the library document has been used on the local area network. See the IsPublishedForLAN property to determine whether or not the library document is available for the LAN.

**TimesUsedOnWeb**
Returns the number of times the library document has been used by RightFax WebTools. See the IsPublishedForWeb property to determine whether or not the library document is available for use on the Web.

**WillExpire**
This Boolean property indicates whether or not the library document has an expiration date. See also the ExpireDate property.

**Methods**

**AdjustUsage**
This method changes the authorized use of the library document. The default is to allow access only on the local area network. Specify one of the valid usage types. You can repeat this command as necessary to allow additional uses for each document.
Table 6.10  LibraryDocumentUsage Enumerated Values

<table>
<thead>
<tr>
<th>Name</th>
<th>Numeric equivalent</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>lduNone</td>
<td>0</td>
<td>No authorized usage.</td>
</tr>
<tr>
<td>lduFOD</td>
<td>1</td>
<td>Allow usage by the RightFax Docs-on-Demand module.</td>
</tr>
<tr>
<td>lduWeb</td>
<td>2</td>
<td>Allow usage on the Web.</td>
</tr>
<tr>
<td>lduLAN</td>
<td>3</td>
<td>This is the default.</td>
</tr>
<tr>
<td>lduResetAll</td>
<td>0x0100</td>
<td>Re-sets usage to None.</td>
</tr>
<tr>
<td>lduResetLastUsedDate</td>
<td>0x0200</td>
<td>See also the LastUsedDate property.</td>
</tr>
</tbody>
</table>

For example, to allow the current library document to be accessed by the RightFax Docs-on-Demand module, enter:

```vbnet
objLibraryDocument.AdjustUsage 1duFOD
```

To re-set the last used date property, you must combine the re-set with the usage type. For example, to change the last used date (and make it appear that the document has not been used on the LAN), enter:

```vbnet
objLibraryDocument.AdjustUsage 1duLAN Or 1duResetLastUsedDate
```

**Delete**

Removes the object from the server.

**Save**

Writes the object to the server.

**Example**

The following example sends a library document as part of the outgoing fax:

```vbnet
If 1stLibDocs.Text = "" Then
    oFax.Send
Else
    libdoc = 1stLibDocs.Text
    oFax.Attachment.Add
    MyFaxServer.LibraryDocuments(libdoc)
    oFax.Send
End If
```
LibraryDocuments Collection

A LibraryDocuments collection is a set of one or more LibraryDocument objects.

Properties

Count
Returns the number of objects in the collection.

Create
Returns a new LibraryDocument object. Note that the user must be authorized to
create new library documents by having the
IsAbleToCreateNewLibraryDocuments property set to True or 1 in the user
object.

Item
Returns the library document object with the specified item number. Each object
in the collection is numbered, starting with 1. See also the Num property.

Methods

Add
Adds an object to the collection.

Refresh
Re-loads the list of library document objects from the fax server.

Remove
Removes a specified object from the collection but not from the server. To
remove an object from the server, use the Delete method in the object you want
to delete.

RemoveAll
Removes all objects from the collection but not from the server. To remove an
object from the server, use the Delete method in the object you want to delete.
Exercise 2

In this exercise you will add to your sample application the ability to send a library document with your fax and overlay a form.

1. Add the ability to send a library document with your fax.
   • Create a list box in your sample application to display a list of library documents available on the system.
   • Add the functionality to your application to select a library document from the list box you created and attach it to a fax.

2. Add the ability to overlay a form to your fax.
   • Create a list box in your sample application to display a list of forms available on the system.
   • Add the functionality to your application to select a form to overlay on your fax.
Section 7
Users, Groups, and Folders

Objectives

Upon completion of this section, you should be able to:

- Utilize the user objects.
- Describe the properties, methods and events associated with Delegates.
- Describe the properties, methods and events associated with Folders.
- Describe the properties, methods and events associated with Groups.
- Describe the properties, methods and events associated with Signatures.
- Describe the properties, methods and events associated with Users.
User Object Overview

You must create a User object for each person who will use the RightFax system. Users can be organized into Groups, which simplifies tasks like defining maintenance defaults. Each user can sort their faxes by creating folders.

Delegate

A delegate is a person who has access to another person’s (the delegator’s) user information. A Delegate object contains one User object. To grant access to multiple users, create a Delegate object for each user and then create a Delegates collection object. The User object for which permission is being granted specifies this Delegates collection in the AccessibleByTheseUsers property.

Properties

- **DeletePermissionCanDeleteFaxes**
  This Boolean property specifies whether or not the delegate is authorized to delete faxes.

- **DeletePermissionCanDeleteFolders**
  This Boolean property specifies whether or not the delegate is authorized to delete folders.

- **DeletePermissionCanDeletePhoneItems**
  This Boolean property specifies whether or not the delegate is authorized to delete phonebook entries.

- **Handle**
  Returns the code used internally by the RightFax system to identify each unique object.

- **MiscPermissionCanApproveFaxes**
  This Boolean property specifies whether or not the delegate is authorized to approve faxes on behalf of the delegator.

- **MiscPermissionCanModifyDelegates**
  This Boolean property specifies whether or not the delegate is authorized to make add or remove users from the delegator’s list of delegates, and to edit the access rights granted to each delegate.

- **MiscPermissionCanToggleRefuseFaxDistribution**
  This Boolean property specifies whether or not the delegate is authorized to change the Refuse Fax Distribution option.
**MiscPermissionFaxRequiresApproval**
This Boolean property specifies whether or not faxes created by the delegate in the delegator’s mailbox will require approval. Approval can only be granted by the delegator, another delegate with permission to approve faxes, or an administrator. True or 1 indicates that the faxes must be approved; False or 0 indicates that faxes will be sent without approval.

**ReadPermissionCanBrowseAllFolders**
This Boolean property specifies whether or not the delegate is authorized to view folders other than the Inbox.

**ReadPermissionCanExportImages**
This Boolean property specifies whether or not the delegate is authorized to save a copy of the fax image from the RightFax fax viewer.

**ReadPermissionCanMailImages**
This Boolean property specifies whether or not the delegate is authorized to send a copy of the fax image to an e-mail address from the RightFax fax viewer.

**ReadPermissionCanPrintFaxes**
This Boolean property specifies whether or not the delegate is authorized to print faxes.

**ReadPermissionCanViewFax**
This Boolean property specifies whether or not the delegate is authorized to view faxes (including the cover page, all body pages, and attachments).

**ReadPermissionCanViewFirstPage**
This Boolean property specifies whether or not the delegate is authorized to view the first page of faxes. When this property is set to True or 1, and the ReadPermissionCanViewFax permission is set to False or 0, the delegate will be able to read only the first page of faxes.

**ReadPermissionCanViewHistory**
This Boolean property specifies whether or not the delegate is authorized to view fax histories.

**ReadPermissionCanUsePhoneEntries**
This Boolean property specifies whether or not the delegate has access to all the delegator’s phonebook entries. True or 1 indicates that the delegate can use all phonebook entries; False or 0 indicates that the delegate can only access the delegator’s shared phonebook entries.

**User**
Specifies the user to whom access is being granted. You can only enter one User object.
**WritePermissionCanAnnotateFaxes**
This Boolean property specifies whether or not the delegate is authorized to annotate faxes.

**WritePermissionCanCreateFaxes**
This Boolean property specifies whether or not the delegate is authorized to create and send new faxes from the delegator’s mailbox.

**WritePermissionCanCreateFolders**
This Boolean property specifies whether or not the delegate is authorized to create new folders.

**WritePermissionCanCreatePhoneItems**
This Boolean property specifies whether or not the delegate is authorized to add new entries to the delegator’s RightFax fax phonebook.

**WritePermissionCanEditFaxes**
This Boolean property specifies whether or not the delegate is authorized to edit a fax’s addressing information after it has been initially sent.

**WritePermissionCanEditPhoneItems**
This Boolean property specifies whether or not the delegate is authorized to edit existing entries in the delegator’s RightFax fax phonebook.

**WritePermissionCanMoveFaxes**
This Boolean property specifies whether or not the delegate is authorized to move faxes between folders in the delegator’s mailbox.

**WritePermissionCanForwardRouteFaxes**
This Boolean property specifies whether or not the delegate is authorized to forward or route faxes.

**WritePermissionCanOCRFaxes**
This Boolean property specifies whether or not the delegate is authorized to use the RightFax OCR Converter software to create text documents from the delegator’s fax images.

**WritePermissionCanRenameFolders**
This Boolean property specifies whether or not the delegate is authorized to rename folders in the delegator’s mailbox.

**WritePermissionCanUpdateFaxStatus**
This Boolean property specifies whether or not the delegate is authorized to update the fax status (retry sending the fax after transmission errors are encountered).
Methods

Delete
Removes the object from the server.

Save
Writes the object to the server.

Delegates Collection

A Delegates collection is a set of one or more Delegate objects. To assign a Delegates collection to a User object, use the AccessibleByTheseUsers property in the User object for which access is being granted.

Properties

Count
Returns the number of objects in the collection.

Create
Returns a new object.

Item
Returns the form object with the specified item number. Each object in the collection is numbered, starting with 1.

Methods

Add
Adds an object to the collection

Refresh
Re-loads the list of objects from the fax server.

Remove
Removes a specified object from the collection but not from the server. To remove an object from the server, use the Delete method in the object you want to delete.

RemoveAll
Removes all objects from the collection but not from the server. To remove an object from the server, use the Delete method in the object you want to delete.
Folder

A Folder is an organizational tool that helps each user sort and categorize faxes. For example, a user can set up an Incoming and an Outgoing folder, or one folder for each of several projects. The Folder object contains the information that identifies the faxes that the user assigns to the folder.

Properties

**ByteID**
Returns the unique identifier of the folder used for database storage.

**Faxes**
Returns the Faxes collection associated with this folder. See the Folders Collection.

**ID**
Specifies a description of the folder. You must specify a folder ID in order to identify the folder later.

Methods

**Delete**
Removes the object from the server.

Folders Collection

A Folders collection is a set of one or more Folder objects.

Properties

**Count**
Returns the number of objects in the collection.

**Create**
Returns a new object.

**Item**
Returns the folder object with the specified item number. Each object in the collection is numbered, starting with 1.
Methods

Add
Adds an object to the collection

Refresh
Re-loads the list of objects from the fax server.

Remove
Removes a specified object from the collection but not from the server. To remove an object from the server, use the Delete method in the object you want to delete.

RemoveAll
Removes all objects from the collection but not from the server. To remove an object from the server, use the Delete method in the object you want to delete.

Group

A group object contains administrative information that is applied to all of the users in the group. Once a group is created, you can assign any Users collection to the group.

Properties

AlternateAdministrator
Specifies the User object of the user who is the alternate administrator for this group. See also PrimaryAdministrator.

AlternateAdministratorID
Specifies the ID of the user who is the alternate administrator for this group. See also PrimaryAdministratorID.

ChangeTag
Returns a number that is incremented each time the object is updated. You can check this number to determine whether or not the group has been updated without getting the entire Group object.

ConcurrentDelayTime
Specifies the start time (in HHMM format) that the RightFax system will send delayed faxes. This parameter is only effective if forced scheduling is enabled. Use the IsForcedSchedulingEnabled property to enable or disable forced scheduling. For example, if you’ve set a limit of 50 faxes that a group can send between 8 A.M. and 5 P.M., any faxes over that limit will be sent beginning at 10 P.M. The delay time would be entered as 2000. See also the MaxConcurrentFaxes and MaxConcurrentPages properties.
ConcurrentEndTime
Specifies the end time (in HHMM format) that concurrent faxes will be limited by forced scheduling. Any faxes in the queue but not yet sent at the specified time will be sent after the specified ConcurrentDelayTime. Use the IsForcedSchedulingEnabled property to enable or disable forced scheduling. For example, if there is a limit of 50 faxes that a group can send between 8 A.M. and 5 P.M., the end time would be entered as 1700.

ConcurrentStartTime
Specifies the start time (in HHMM format) that concurrent faxes will be limited by forced scheduling. Use the IsForcedSchedulingEnabled property to enable or disable forced scheduling. For example, if there is a limit of 50 faxes that a group can send between 8 A.M. and 5 P.M., the start time would be entered as 0800.

CoverSheet
The name of the default cover sheet. This can be changed by each user with the CoverSheetModelID property or for each fax with the FCSFilename property.

FaxAgeCertifyDelivery
Specifies the number days a fax sent via certified delivery will remain on the system before being permanently removed. To leave certified delivery faxes on the system indefinitely, set this property to 0.

FaxAgeDeleted
Specifies the number days a fax marked for deletion will remain on the system before being permanently removed. To leave deleted faxes on the system indefinitely, set this property to 0. The default is 30 days. See also ExcludedFromFaxAging in the User object.

FaxAgeFailed
Specifies the number of days a failed fax will remain on the system before being deleted. To leave failed faxes on the system indefinitely, set this property to 0. See also ExcludedFromFaxAging in the User object.

FaxAgeIncomplete
Specifies the number of days an incomplete fax will remain on the system before being deleted. To leave incomplete faxes on the system indefinitely, set this property to 0. See also ExcludedFromFaxAging in the User object.

FaxAgeNew
Specifies the number of days a new fax will remain on the system before being deleted. To leave new faxes on the system indefinitely, set this property to 0. See also ExcludedFromFaxAging in the User object.
**FaxAgeReceived**
Specifies the number of days a received fax will remain on the system before being deleted. To leave received faxes on the system indefinitely, set this property to 0. See also ExcludedFromFaxAging in the User object.

**FaxAgeSent**
Specifies the number of days a sent fax will remain on the system before being deleted. To leave sent faxes on the system indefinitely, set this property to 0. See also ExcludedFromFaxAging in the User object.

**FaxAgeSubFolders**
This Boolean value determines whether or not fax aging will be performed on all user’s folders. True or 1 indicates that all users’ subfolders will be aged; False or 0 indicates that only the Main and Trash folders will be aged.

**GroupRoutingCode**
Specifies the default routing code for the group if RightFax SmartFax Distribution is enabled. Set this property to –1 to disable SmartFax Distribution.

**GroupSFDEnabled**
This Boolean property indicates whether or not RightFax SmartFax Distribution is enabled. True or 1 indicates that it is enabled; False or 0 indicates that it is not.

**GroupSFDLoginRequired**
This Boolean property indicates whether or not the user is required to enter a password to use the RightFax Smart Fax Distribution feature. True or 1 indicates that a password is required; False or 0 indicates that it is not required. See also the GroupSFDType and GroupSFDEnabled properties.

**GroupSFDType**
Specifies how faxes should be distributed if the RightFax Smart Fax Distribution feature is enabled. You must specify one of the valid GroupSFDType values.

### Table 7.1  GroupSFDType Enumerated Values

<table>
<thead>
<tr>
<th>Name</th>
<th>Numeric equivalent</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>gsfdLinear</td>
<td>0</td>
<td>Distributes to the users in order regardless of number of faxes.</td>
</tr>
<tr>
<td>gsfdBalanced</td>
<td>1</td>
<td>Distributes first to the user who has the fewest faxes in the fax inbox.</td>
</tr>
</tbody>
</table>
Handle
Returns the code used internally by the RightFax system to identify each unique object.

HideUIControlAddPhoneBookEntryButton
This Boolean value indicates whether or not to display the Add Entry button on the Fax Information dialog box when the members of this group create new faxes.

HideUIControlAlternateFaxNumber
This Boolean value indicates whether or not to display the Alt Fax Number box on the Fax Information dialog box when the members of this group create new faxes.

HideUIControlAutoDeleteSetting
This Boolean value indicates whether or not to display the Automatic Deletion box on the Fax Information dialog box when the members of this group create new faxes.

HideUIControlBillInfo1
This Boolean value indicates whether or not to display the first billing code box on the Fax Information dialog box when the members of this group create new faxes.

HideUIControlBillInfo2
This Boolean value indicates whether or not to display the second billing code box on the Fax Information dialog box when the members of this group create new faxes.

HideUIControlBillInfoLookup
This Boolean value indicates whether or not to display the Billing Code Lookup button on the Fax Information dialog box when the members of this group create new faxes.

HideUIControlBodyAttachments
This Boolean value indicates whether or not to display the Alternate Body Attachment checkbox on the Fax Information dialog box when the members of this group create new faxes.

HideUIControlCallBack
This Boolean value indicates whether or not to display the CallBack checkbox on the Fax Information dialog box when the members of this group create new faxes.
HideUIControlCertified
This Boolean value indicates whether or not to display the Certified Delivery checkbox on the Fax Information dialog box when the members of this group create new faxes.

HideUIControlCity
This Boolean value indicates whether or not to display the To City/State checkbox on the Fax Information dialog box when the members of this group create new faxes.

HideUIControlCompany
This Boolean value indicates whether or not to display the To Company box on the Fax Information dialog box when the members of this group create new faxes.

HideUIControlCoverSheetSelect
This Boolean value indicates whether or not to display the Cover Sheet File box on the Fax Information dialog box when the members of this group create new faxes.

HideUIControlCoverSheetToggle
This Boolean value indicates whether or not to display the Use cover sheet checkbox on the Fax Information dialog box when the members of this group create new faxes.

HideUIControlDelaySend
This Boolean value indicates whether or not to display the Delay send checkbox on the Fax Information dialog box when the members of this group create new faxes.

HideUIControlFormType
This Boolean value indicates whether or not to display the Use form boxes on the Fax Information dialog box when the members of this group create new faxes.

HideUIControlFromName
This Boolean value indicates whether or not to display the From Name box on the Fax Information dialog box when the members of this group create new faxes.

HideUIControlFromPhoneNumber
This Boolean value indicates whether or not to display the From Voice Number box on the Fax Information dialog box when the members of this group create new faxes.
HideUIControlGeneralFax
This Boolean value indicates whether or not to display the From Company fax number box on the Fax Information dialog box when the members of this group create new faxes.

HideUIControlGeneralVoice
This Boolean value indicates whether or not to display the From Company voice number box on the Fax Information dialog box when the members of this group create new faxes.

HideUIControlLibraryDocumentSelect
This Boolean value indicates whether or not to display the list of available library document attachments on the Fax Information dialog box when the members of this group create new faxes.

HideUIControlMoreButton
This Boolean value indicates whether or not to display the More Options tab on the Fax Information dialog box when the members of this group create new faxes.

HideUIControlNativeDocumentAttachments
This Boolean value indicates whether or not to display the Native Document Attachment checkbox on the Fax Information dialog box when the members of this group create new faxes.

HideUIControlNotesButton
This Boolean value indicates whether or not to display the Cover Sheet Notes tab on the Fax Information dialog box when the members of this group create new faxes.

HideUIControlPDFOptions
This Boolean value indicates whether or not to display the PDF Options button on the Fax Information dialog box when the members of this group create new faxes.

HideUIControlPhoneBookButton
This Boolean value indicates whether or not to display the Phonebook button on the Fax Information dialog box when the members of this group create new faxes.

HideUIControlPreview
This Boolean value indicates whether or not to display the Hold for preview checkbox on the Fax Information dialog box when the members of this group create new faxes.
HideUIControlPriority
This Boolean value indicates whether or not to display the Priority box on the Fax Information dialog box when the members of this group create new faxes.

HideUIControlPrivateFax
This Boolean value indicates whether or not to display the From Private fax number box on the Fax Information dialog box when the members of this group create new faxes.

HideUIControlRecipientType
This Boolean value indicates whether or not to display the recipient type (fax or e-mail) selection button on the Fax Information dialog box when the members of this group create new faxes.

HideUIControlSecureSend
This Boolean value indicates whether or not to display the Recipient Fax ID box on the Fax Information dialog box when the members of this group create new faxes.

HideUIControlSmartResume
This Boolean value indicates whether or not to display the Use smart resume checkbox on the Fax Information dialog box when the members of this group create new faxes.

HideUIControlTransmitQuality
This Boolean value indicates whether or not to display the Fine mode checkbox on the Fax Information dialog box when the members of this group create new faxes.

HideUIControlUsePDF
This Boolean value indicates whether or not to display the Create PDF image checkbox on the Fax Information dialog box when the members of this group create new faxes.

HideUIControlVoiceNumber
This Boolean value indicates whether or not to display the To Voice number box on the Fax Information dialog box when the members of this group create new faxes.

ID
Specifies an ID you can assign to identify each unique object. If you specify a value without specifying a property name, the system will assign that value as the object ID.
Users, Groups, and Folders

**IsForcedSchedulingEnabled**
Specifies whether or not forced scheduling is enabled. This feature allows you to automatically hold all outgoing faxes in a queue until a specified time, such as late night when transmission traffic might be less. Use the ConcurrentStartTime property to specify when the faxes will begin transmitting; use the ConcurrentEndTime property to specify when forced scheduling should end.

**MaxConcurrentFaxes**
If forced scheduling is enabled, this specifies the maximum number of faxes that can be scheduled by the users in this group for transmission between the specified start and end times. Use the IsForcedSchedulingEnabled property to enable or disable forced scheduling. Use the ConcurrentStartTime property to specify when the faxes will begin transmitting; use the ConcurrentEndTime property to specify when forced scheduling should end.

**MaxConcurrentPages**
If forced scheduling is enabled, this specifies the maximum number of pages that can be scheduled by the users in this group for transmission between the specified start and end times. Use the IsForcedSchedulingEnabled property to enable or disable forced scheduling. Use the ConcurrentStartTime property to specify when the faxes will begin transmitting; use the ConcurrentEndTime property to specify when forced scheduling should end.

**MustHaveCoversheet**
This Boolean property specifies whether or not the users in this group are required to use a cover sheet when sending a fax. True or 1 indicates the users must have a cover sheet; False or 0 indicates they are not required to have a cover sheet. If the users are not required to have a cover sheet with every fax, users can specify their own default with the IsCoverPageDefaultedOn property, or can specify whether or not each fax has a cover page with the HasCoversheet property.

**MustHoldForPreview**
This Boolean property specifies whether or not all faxes sent by the users in this group will be held for preview.

**NotifyType**
Specifies the default method by which a user in the group will receive notification of a received fax. This can be changed by each user with the NotifyType property in the User object.
Table 7.2  NotifyType Enumerated Values

<table>
<thead>
<tr>
<th>Name</th>
<th>Numeric equivalent</th>
</tr>
</thead>
<tbody>
<tr>
<td>nNetworkBroadcast</td>
<td>0</td>
</tr>
<tr>
<td>nCustom1</td>
<td>1</td>
</tr>
<tr>
<td>nCustom2</td>
<td>2</td>
</tr>
<tr>
<td>nCustom3</td>
<td>3</td>
</tr>
<tr>
<td>nCustom4</td>
<td>4</td>
</tr>
<tr>
<td>nCustom5</td>
<td>5</td>
</tr>
<tr>
<td>nCustom6</td>
<td>6</td>
</tr>
<tr>
<td>nCustom7</td>
<td>7</td>
</tr>
<tr>
<td>nCustom8</td>
<td>8</td>
</tr>
<tr>
<td>nCustom9</td>
<td>9</td>
</tr>
<tr>
<td>nCCMail</td>
<td>10</td>
</tr>
<tr>
<td>nMSMail</td>
<td>11</td>
</tr>
<tr>
<td>nGroupWise</td>
<td>12</td>
</tr>
<tr>
<td>nTRS</td>
<td>13</td>
</tr>
<tr>
<td>nCX3</td>
<td>14</td>
</tr>
<tr>
<td>nPager</td>
<td>15</td>
</tr>
<tr>
<td>nNotes</td>
<td>16</td>
</tr>
<tr>
<td>nExchange</td>
<td>17</td>
</tr>
<tr>
<td>nSMTP</td>
<td>18</td>
</tr>
</tbody>
</table>

**NumberOfMembers**  
Specifies the number of members in the group. See also the Users property to specify the User collection.

**PrimaryAdministrator**  
Specifies the User object of the user who is the primary administrator for this group. See also AlternateAdministrator.

**PrimaryAdministratorID**  
Specifies the ID of the user who is the alternate administrator for this group. See also AlternateAdministratorID.

**Users**  
Specifies the Users Collection that contains the users who are in this group.
Users, Groups, and Folders

Methods

Delete
Removes the group object from the server. You can only delete groups that do not have any members.

Save
Writes the object to the server.

Example

The following example creates a group called “support” and automatically deletes sent and received faxes for the group after 90 days.

```vbscript
Set obGroup = MyFaxServer.CreateObject(coGroup)
obGroup.ID = “support”
obGroup.FaxAgeReceived = 90
obGroup.FaxAgeSent = 90
```

Groups Collection

A Groups collection is a set of one or more Group objects.

Properties

Count
Returns the number of objects in the collection.

Create
Returns a new object.

Item
Returns the object with the specified item number. Each object in the collection is numbered, starting with 1.
**Methods**

**Add**
Adds an object to the collection

**Refresh**
Re-loads the list of group objects from the fax server.

**Remove**
Removes a specified object from the collection but not from the server. To remove an object from the server, use the Delete method in the object you want to delete.

**RemoveAll**
Removes all objects from the collection but not from the server. To remove an object from the server, use the Delete method in the object you want to delete.

**Signature**
A signature is stored as an encrypted image file on the fax server. The Signature object contains the name of the file and a set of up to three users who are authorized to use the signature.

**Properties**

**Description**
Specifies the description of the signature object.

**FileName**
Specifies the name of the signature image file.

**Handle**
Returns the code used internally by the RightFax system to identify each unique object.

**ID**
Specifies an ID you can assign to identify each unique object. If you specify a value without specifying a property name, the system will assign that value as the object ID.

**Owner**
Specifies the user who is the owner of the signature.

**Users**
Specifies the Users Collection that identifies the users who are authorized to use the Signature object. There can be up to three users in the collection.
Methods

Delete
Removes the object from the server.

Refresh
Re-loads the list of signature objects from the fax server.

Save
Writes the object to the server.

Signatures Collection

A Signatures collection is a set of one or more Signature objects.

Properties

Count
Returns the number of objects in the collection.

Create
Returns a new object.

Item
Returns the object with the specified item number. Each object in the collection is numbered, starting with 1.

Methods

Add
Adds an object to the collection

Refresh
Re-loads the list of signature objects from the fax server.

Remove
Removes a specified object from the collection but not from the server. To remove an object from the server, use the Delete method in the object you want to delete.

RemoveAll
Removes all objects from the collection but not from the server. To remove an object from the server, use the Delete method in the object you want to delete.
User

A user object contains all the information about a specific person who will have access to the RightFax system. This includes authorizations for RightFax modules and defaults for user preferences that can be changed for each fax.

Properties

**AccessibleByTheseUsers**
Specifies a Delegates Collection that will have access to this User object, including associated faxes and attachments.

**AlternateNotifyUser**
Specifies a string value for the user ID of the user object to whom notifications will be sent if the current user has the alternate notification feature turned on.

**AlternateNotifyUserID**
Specifies the User object to whom notifications will be sent if the current user has the alternate notification feature turned on. See also the IsAlternateNotificationEnabled property.

**AreReceivedFaxesStamped**
This Boolean property indicates whether or not incoming faxes will be marked with the time and date that the fax was received. True or 1 indicates the incoming faxes will be stamped; False or 0 indicates they will not.

**AssociatedNTAccount**
Specifies a string value the Windows NT domain and account associated with the user in the format Domain/Account. See also IsNTAuthenticationRequired.

**AssociatedNTAccountSID**
The user’s Windows NT SID in binary form. See also IsNTAuthenticationRequired.

**AutoforwardFaxNumber**
Specifies the fax number to which incoming faxes will be forwarded. The IsAutoforwardEnabled property must be set to True or 1 and the AutoForwardType must be set to afFaxNumber or 0 for the fax to be forwarded to this number.

**AutoForwardType**
Specifies how incoming faxes will be forwarded. The IsAutoforwardEnabled property must be set to True or 1 for automatic forwarding to take place. This property requires one of the following enumerated values.
### Table 7.3 AutoForwardType Enumerated Values

<table>
<thead>
<tr>
<th>Name</th>
<th>Numeric equivalent</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>afFaxNumber</td>
<td>0</td>
<td>See also AutoforwardFaxNumber</td>
</tr>
<tr>
<td>afUser</td>
<td>1</td>
<td>None</td>
</tr>
</tbody>
</table>

**AutoOCRExtension**

Specifies the three-letter extension that will be appended to the file name when an incoming fax is processed with optical character recognition algorithms. For example, the extension might be `.doc` or `.txt`. IsAutoOCREnabled must be set to True or 1 for this to take effect.

**AutoOCRFormat**

Specifies the default output format for the resulting file when an incoming fax is processed with optical character recognition algorithms. You must specify one of the valid format types. The user can change this for a specific fax with the OCR method of the CoverSheets Collection. IsAutoOCREnabled must be set to True or 1 for this to take effect.

### Table 7.4 OCRFormatType Enumerated Values

<table>
<thead>
<tr>
<th>Name</th>
<th>Numeric equivalent</th>
</tr>
</thead>
<tbody>
<tr>
<td>ofSmartASCII</td>
<td>0</td>
</tr>
<tr>
<td>ofASCII</td>
<td>1</td>
</tr>
<tr>
<td>ofRTF</td>
<td>3</td>
</tr>
</tbody>
</table>

**AutoOCRLayout**

Specifies the output layout for the resulting file when an incoming fax is processed with optical character recognition algorithms. You must specify one of the valid layout types. The user can override this for a specific fax with the OCR method of the CoverSheets Collection. IsAutoOCREnabled must be set to True or 1 for this to take effect.
Table 7.5 OCRLAYOUTTYPE Enumerated Values

<table>
<thead>
<tr>
<th>Name</th>
<th>Numeric equivalent</th>
</tr>
</thead>
<tbody>
<tr>
<td>OLWYSIWYG</td>
<td>0</td>
</tr>
<tr>
<td>OLLEFTJUSTIFIED</td>
<td>1</td>
</tr>
</tbody>
</table>

**ChangeTag**
Returns a number that is incremented each time the object is updated. You can check this number to determine whether or not the user has been updated without getting the entire User object.

**CheckForNewFaxes**
Returns a new fax object, if any. Repeat this property as needed. For example:

```vba
‘Timer_Timer gets fired by the form every x milliseconds
Private Sub Timer_Timer()

RestartLoop:
    On Error GoTo LoopExit

    Set newFax = curUser.CheckForNewFaxes
    ’ do something here with newFax
    GoTo RestartLoop

LoopExit:
End Sub
```

**CoverSheetModelID**
Specifies the file name of the default cover sheet for the user. This can also be set at the group level with the CoverSheet property, or it can be set for each fax with the FCSFilename property.

**DefaultBillingCode**
Specifies the BillingCode object that will be used as a default for all faxes relating to this user. This can be changed for each fax with the BillingCode property.

**DefaultBillingCode1**
Specifies a string value for the first default billing code for the user.

**DefaultBillingCode2**
Specifies a string value for the second default billing code for the user.

**DefaultFromName**
Specifies the default sender name that will be used in the cover sheet. This can be changed for each fax with the FromName property.
**DefaultOCRFileExtension**
Specifies the default file extension for the file generated when the fax is processed through the optical character recognition software. This can be changed for each fax by the OCR method. See also the AutoOCRExtension property.

**DefaultOCRFormat**
Specifies the default format for the file generated when the fax is processed through the optical character recognition software. This can be changed for each fax by the OCR method. See also the AutoOCRFormat property.

<table>
<thead>
<tr>
<th>Name</th>
<th>Numeric equivalent</th>
</tr>
</thead>
<tbody>
<tr>
<td>ofSmartASCII</td>
<td>0</td>
</tr>
<tr>
<td>ofASCII</td>
<td>1</td>
</tr>
<tr>
<td>ofRTF</td>
<td>2</td>
</tr>
</tbody>
</table>

**DefaultOCRLAYOUT**
Specifies the default layout for the file generated when the fax is processed through the optical character recognition software. You must specify a valid OCRLAYOUTType value. This can be changed for each fax by the OCR property. See also the AutoOCRLAYOUT property.

<table>
<thead>
<tr>
<th>Name</th>
<th>Numeric equivalent</th>
</tr>
</thead>
<tbody>
<tr>
<td>oIWYSIWYG</td>
<td>0</td>
</tr>
<tr>
<td>oIleftJustified</td>
<td>1</td>
</tr>
</tbody>
</table>

**DefaultPrinter**
Specifies the Printer object where faxes will print when the user specifies the print function. To specify the printer for automatic fax printing, see the SentAutoprintPrinter and ReceivedAutoprintPrinter properties.

**DefaultPrinterID**
Specifies a string value for the printer ID where faxes will print when the user specifies the print function. To specify the printer for automatic fax printing, see the SentAutoprintPrinter and ReceivedAutoprintPrinter properties.
DefaultPriority
Specifies the default priority for the fax using one of the valid FaxPriorityType enumerated values. This can be changed for each fax with the Priority property. Note that in order for the user to send a fax with High priority, the IsAbleToUseHighPriority property must be set to set to True or 1, and the HighestAvailablePriority property must be set to fpHigh.

Table 7.8 FaxPriorityType Enumerated Values

<table>
<thead>
<tr>
<th>Name</th>
<th>Numeric equivalent</th>
</tr>
</thead>
<tbody>
<tr>
<td>fpNormal</td>
<td>0</td>
</tr>
<tr>
<td>fpLow</td>
<td>1</td>
</tr>
<tr>
<td>fpHigh</td>
<td>2</td>
</tr>
</tbody>
</table>

DefaultToCityState
Specifies the default recipient city and state information. This can be changed for each fax with the ToCityState property.

DefaultToCompany
Specifies the default recipient company information. This can be changed for each fax with the ToCompany property.

DefaultToFaxNumber
Specifies the default fax number that will be entered for received faxes. This can be changed for each fax with the ToFaxNumber property.

DefaultToName
Specifies the default recipient name. This can be changed for each fax with the ToName property.

DefaultToVoiceNumber
Specifies the default recipient’s telephone number, if any. This can be changed for each fax with the ToVoiceNumber property.

DeleteAfterSuccessfulRoute
This Boolean property specifies whether or not the fax will be removed from the user’s fax list after it has been routed to someone else. True or 1 indicates the fax will be removed; False or 0 indicates it will not.

DIDNumber
Specifies the user’s direct inward dialing number.
**DistinguishedName**
Specifies a fully qualified name for certain applications.

**EmailAddress**
Specifies the user’s return e-mail address. This may be required by SMTP servers when sending documents to e-mail addresses.

**EmailRouteForm**
Specifies the type of form used to route a fax to the e-mail system. You must specify one of the valid EmailRouteFormType values.

<table>
<thead>
<tr>
<th>Name</th>
<th>Numeric equivalent</th>
</tr>
</thead>
<tbody>
<tr>
<td>erfDefault</td>
<td>0</td>
</tr>
<tr>
<td>erfNone</td>
<td>1</td>
</tr>
<tr>
<td>erfStandard</td>
<td>2</td>
</tr>
<tr>
<td>erfAdvanced</td>
<td>3</td>
</tr>
</tbody>
</table>

**ExcludedFromFaxAging**
This Boolean property indicates whether or not this user’s faxes will be deleted after a specified time. True or 1 indicates the faxes will not be deleted due to aging; False or 0 indicates that the faxes will be deleted due to aging. See also FaxAgeCertifyDelivery, FaxAgeFailed, FaxAgeNew, FaxAgeReceived, and FaxAgeSent, all of which are set in the Group object.

**Faxes**
Returns the Faxes Collection for the user. See also the CoverSheet object.

**Folders**
Returns the Folders Collection for the user. See also the Folder object.

**GeneralFaxNumber**
Specifies the general fax number, such as for a department.

**GeneralVoiceNumber**
Specifies the general telephone number, such as for a department or switchboard.

**Handle**
Returns the code used internally by the RightFax system to identify each unique object.
HasAccessToTheseUsers
Returns the Delegates Collection that contains the list of users who have designated this user as a delegate. See also the Delegate object.

HighestAvailablePriority
Specifies the highest priority available to this user. The priority must be one of the valid FaxPriorityType values. Note that in order for the user to send a fax with High priority, this property must be set to fpHigh and the IsAbleToUseHighPriority properties must be set to True or 1. See also the DefaultPriority and the Priority property.

Table 7.10 FaxPriorityType Enumerated Values

<table>
<thead>
<tr>
<th>Enumerated value</th>
<th>Numeric equivalent</th>
</tr>
</thead>
<tbody>
<tr>
<td>fpNormal</td>
<td>0</td>
</tr>
<tr>
<td>fpLow</td>
<td>1</td>
</tr>
<tr>
<td>fpHigh</td>
<td>2</td>
</tr>
</tbody>
</table>

ID
Specifies an ID you can assign to identify each unique object. If you specify a value without specifying a property name, the system will assign that value as the object ID.

IsAbleToChangeCoverSheets
This Boolean property indicates whether or not the user is authorized to modify the cover sheets. True or 1 indicates that the user is authorized to make changes; False or 0 indicates that the user is not authorized to make changes.

IsAbleToCreateNewForms
This Boolean property indicates whether or not the user is authorized to create a new Form object. True or 1 indicates that the user is authorized to create new forms; False or 0 indicates that the user is not authorized to create new forms. See also the Form object.

IsAbleToCreateNewLibraryDocuments
This Boolean property indicates whether or not the user is authorized to create a new LibraryDocument object. True or 1 indicates that the user is authorized to create new library documents; False or 0 indicates that the user is not authorized to create new library documents. See also the LibraryDocument object.
**IsAbleToOCR**  
This Boolean property specifies whether or not the user is authorized to use the RightFax OCR Converter software to create text documents from fax images. True or 1 indicates the user is authorized; False or 0 indicates the user is not authorized.

**IsAbleToRunReports**  
This Boolean property specifies whether or not the user is authorized to run reports. True or 1 indicates the user is authorized; False or 0 indicates the user is not authorized.

**IsAbleToUseHighPriority**  
This Boolean property specifies whether or not the user is authorized to send faxes with priority set to High. True or 1 indicates the user is authorized; False or 0 indicates the user is not authorized. Note that in order for the user to send a fax with High priority, this property must be set to set to True or 1 and the HighestAvailablePriority property must be set to fpHigh. See also the DefaultPriority property and the Priority property in the Faxes Collection.

**IsAdministrator**  
This Boolean property indicates whether or not the user has permission to control the fax server settings and features. True or 1 indicates the user is an administrator; False or 0 indicates the user is not an administrator.

**IsAlternateNotificationEnabled**  
This Boolean property indicates whether or not the user has turned on the alternate notification feature. True or 1 indicates this feature is on; False or 0 indicates it is not on. Alternate notification sends a notice to a person other than the user when certain events occur. For example, a user can receive notification when a fax does not transmit properly (by setting SendNotifyOnSendFailure to True or 1). The notice can also be sent to an alternate recipient, such as a support desk. The user must specify the AlternateNotifyUser property.

**IsArchiveEnable**  
This Boolean property specifies whether or not the user is authorized to generate archive events. True or 1 indicates the user is authorized; False or 0 indicates the user is not authorized.

**IsAutoforwardEnabled**  
This Boolean property specifies whether or not incoming faxes for this user should automatically be forwarded to a specified person or fax number. True or 1 indicates the faxes will be forwarded; False or 0 indicates they will not be forwarded. To use this feature, you must specify the AutoForwardType property. If the forwarding type is fax, you must also specify the fax number with the AutoforwardFaxNumber property.
IsAutoOCREnabled
This Boolean property specifies whether or not the incoming faxes for this should automatically be converted to text files with the RightFax OCR Converter software. True or 1 indicates the conversion happens automatically; False or 0 indicates the fax is not automatically converted. See also the AutoOCRExtension, AutoOCRFormat, and AutoOCRLayout properties.

IsAutoprintReceivedBodyEnabled
This Boolean property specifies whether or not the body of received faxes should automatically print on the default printer specified by the ReceivedAutoprintPrinter property. True or 1 indicates the body of the faxes will automatically print; False or 0 indicates the body of the faxes will not automatically print. The IsFaxesSetToBeAutomaticallyPrinted property must be set to True or 1 for this property to be valid.

IsAutoprintReceivedCoversheetEnabled
This Boolean property specifies whether or not the cover sheet of received faxes should automatically print on the default printer specified by the ReceivedAutoprintPrinter property. True or 1 indicates the fax cover sheets will automatically print; False or 0 indicates the fax cover sheets will not automatically print. The IsFaxesSetToBeAutomaticallyPrinted property must be set to True or 1 for this property to be valid.

IsAutoprintReceivedHistoryEnabled
This Boolean property specifies whether or not the history of received faxes should automatically print on the default printer specified by the ReceivedAutoprintPrinter property. True or 1 indicates the history will automatically print; False or 0 indicates the history will not automatically print. The IsFaxesSetToBeAutomaticallyPrinted property must be set to True or 1 for this property to be valid.

IsAutoprintSentBodyEnabled
This Boolean property specifies whether or not the body of sent faxes should automatically print on the default printer specified by the SentAutoprintPrinter property. True or 1 indicates the body of the faxes will automatically print; False or 0 indicates the body of the faxes will not automatically print. The IsAutoprintSentEnabled property must be set to True or 1 for this to take effect. The IsFaxesSetToBeAutomaticallyPrinted property must be set to True or 1 for this property to be valid.
IsAutoprintSentCoversheetEnabled
This Boolean property specifies whether or not the cover sheet of sent faxes should automatically print on the default printer specified by the SentAutoprintPrinter property. True or 1 indicates the fax cover sheets will automatically print; False or 0 indicates the fax cover sheets will not automatically print. The IsAutoprintSentEnabled property must be set to True or 1 for this to take effect. The IsFaxesSetToBeAutomaticallyPrinted property must be set to True or 1 for this property to be valid.

IsAutoprintSentEnabled
This Boolean property specifies whether or not all sent faxes should automatically print on the default printer specified by the SentAutoprintPrinter property. True or 1 indicates the faxes will automatically print; False or 0 indicates the faxes will not automatically print. The IsAutoprintSentEnabled property must be set to True or 1 for this property to be valid.

IsAutoprintSentFailedFaxesEnabled
This Boolean property specifies whether or not faxes that failed during the send process should automatically print on the default printer specified by the SentAutoprintPrinter property. True or 1 indicates the faxes will automatically print; False or 0 indicates the faxes will not automatically print. The IsAutoprintSentEnabled property and the IsFaxesSetToBeAutomaticallyPrinted property must both be set to True or 1 for this to take effect.

IsAutoprintSentHistoryEnabled
This Boolean property specifies whether or not the history of sent faxes should automatically print on the default printer specified by the SentAutoprintPrinter property. True or 1 indicates the history will automatically print; False or 0 indicates the history will not automatically print. The IsAutoprintSentEnabled property and the IsFaxesSetToBeAutomaticallyPrinted property must both be set to True or 1 for this to take effect.

IsAutoprintSentSuccessFaxesEnabled
This Boolean property specifies whether or not faxes that were successfully sent should automatically print on the default printer specified by the SentAutoprintPrinter property. True or 1 indicates the faxes will automatically print; False or 0 indicates the faxes will not automatically print. The IsAutoprintSentEnabled property and the IsFaxesSetToBeAutomaticallyPrinted property must both be set to True or 1 for this to take effect.

IsCallbackRequested
This Boolean property specifies whether or not the user default is to request the recipient to call the voice number when the fax is transmitted. This can be changed for each fax with the IsCallbackRequested property in the Fax object. True or 1 indicates a call is requested; False or 0 indicates it is not.
**IsCoverPageDefaultedOn**
This Boolean property specifies whether or not the default is set to include a cover page with outgoing faxes. True or 1 indicates a cover page will be sent; False or 0 indicates it will not be sent. This can be changed by the HasCoversheet property for each fax.

**IsDefaultCoverSheetsFine**
This Boolean property specifies whether or not the default is set to send the cover page in high resolution (200×200 dots per inch). True or 1 indicates a cover page will be sent in fine mode; False or 0 indicates it will be sent in low resolution (100×100 dots per inch). This can be changed by the IsCoversheetFine property for each fax.

**IsDefaultFine**
This Boolean property specifies whether or not the default is set to send the body of the fax in high resolution (200×200 dots per inch). True or 1 indicates a fax will be sent in fine resolution; False or 0 indicates it will be sent in low resolution (100×100 dots per inch). This can be changed by the IsFineMode property for each fax.

**IsDefaultSmartResumeEnabled**
This Boolean property specifies whether or not the RightFax Smart Resume feature is enabled. This feature stores information about transmitting faxes so that if an interruption occurs, the transmission can be continued from the point of the interruption rather than re-transmitting the entire fax. True or 1 indicates that SmartResume is enabled by default; False or 0 indicates it is not enabled. This can be changed by the UseSmartResume property for each fax.

**IsEmptyTrashEnabled**
This Boolean property specifies whether or not faxes in the trash folder will be deleted when the user exits the system. True or 1 deletes the contents of the trash folder; False or 0 indicates the faxes remain in the trash until it is manually emptied.

**IsExemptFromBillingCodeVerification**
This Boolean property specifies whether or not billing codes submitted by this user will be checked against the list of valid billing codes. True or 1 indicates the codes will not be checked; False or 0 indicates they will be checked. See also the BillingCode object.
IsFaxesSetToBeAutomaticallyPrinted
This Boolean property specifies whether or not faxes will automatically print after transmission is attempted. True or 1 indicates automatic printing; False or 0 indicates that faxes will not automatically be printed. To control which faxes print, see the IsAutoprintReceivedBodyEnabled, IsAutoprintReceivedCoversheetEnabled, IsAutoprintReceivedHistoryEnabled, IsAutoprintSentBodyEnabled, IsAutoprintSentCoversheetEnabled, IsAutoprintSentEnabled, IsAutoprintSentFailedFaxesEnabled, IsAutoprintSentHistoryEnabled, and IsAutoprintSentSuccessFaxesEnabled properties. To specify the printer for automatic printing, use the ReceivedAutoprintPrinter and SentAutoprintPrinter properties.

IsNotAllowedToSearchBillingCodes
This Boolean Property specifies whether or not the user is authorized to search the BillingCodes Collection. True or 1 indicates the user is not allowed to search; False or 0 indicates the user is allowed to search. See also the SearchOnKey and SearchString properties.

IsNTAuthenticationRequired
This Boolean property specifies whether or not authentication of this user is required. True or 1 indicates that authentication is required; False or 0 indicates it is not. If authentication is required, a valid AssociatedNTAccount must exist.

IsPasswordRequired
This Boolean property specifies whether or not this user must enter a password in order to access the RightFax system. True or 1 indicates that a password is required; False or 0 indicates it is not. See also the Password property.

IsRestrictedToFirstPageOnly
This Boolean property specifies whether the user is authorized full access to faxes or is limited to viewing only the first page of each fax. True or 1 indicates the user can only view the first page of each fax; False or 0 indicates the user has full access.

IsSmartFaxDistributionEnabled
This Boolean property specifies whether or not the Smart Fax Distribution feature is turned on. This feature determines the order in which a fax is delivered to multiple recipients. For example, it can be set to send a fax first to the recipient with the fewest faxes in queue. True or 1 indicates this feature is on; False or 0 indicates it is not on. See also GroupSFDEnabled and GroupSFDType in the Group object.

IsSynched
This Boolean property specifies whether or not the user object was created by an automated synchronization process. True or 1 indicates that there is a synchronization process; False or 0 indicates there is not.
Users, Groups, and Folders

**IsTrashEnabled**
This Boolean property specifies whether or not deleted faxes for this user will be stored in the trash folder. True or 1 indicates that deleted faxes will be stored in the trash folder; False or 0 indicates they will be deleted. The faxes in the trash folder remain on the system until the user empties the trash or until the aging time limit is reached. See also the FaxAgeCertifyDelivery property.

**IsUnableToAnnotate**
This Boolean property specifies whether or not the user is allowed to add annotations to received faxes. True or 1 indicates the user is not allowed to annotate; False or 0 indicates the user is allowed to annotate.

**IsUnableToDelete**
This Boolean property specifies whether or not the user is allowed to delete faxes. True or 1 indicates the user is not allowed to delete; False or 0 indicates the user is allowed to delete.

**IsUnableToEditBillingCodes**
This Boolean property specifies whether or not the user is allowed to edit BillingCode objects. True or 1 indicates the user is not allowed to edit billing codes; False or 0 indicates the user is allowed to edit billing codes.

**IsUnprotected**
This Boolean property specifies whether or not the user’s faxes are available for other people to see without entering the user’s password. True or 1 indicates that anyone can browse through this user’s faxes. False or 0 indicates that a password is required.

**IsWebRFXEnabled**
This Boolean property specifies whether or not the graphic images which will be posted to a web site will be in RFX format. True or 1 indicates the graphics will be in RFX format. False or 0 indicates they will not. The default is GIF format. See also the IsWebTiffEnabled property.

**IsWebTiffEnabled**
This Boolean property specifies whether or not the graphic images which will be posted to a web site will be in TIFF format. True or 1 indicates the graphics will be in TIFF format. False or 0 indicates they will not. The default is GIF format. See also the IsWebRFXEnabled property.

**MemberOfGroup**
Specifies the name of the group to which this user belongs. If you change this property, the MemberOfGroupID for this user and the Users property in the Group object also change.
Users, Groups, and Folders

**MemberOfGroupId**
Specifies the ID of the group to which this user belongs. If you change this property, the MemberOfGroup for this user and the Users property in the Group object also change.

**NeedsFaxesApproved**
This Boolean property specifies whether or not faxes sent by this user must go through an approval process. True or 1 indicates that approval is required; False or 0 indicates that approval is not required.

**NotifyType**
Specifies the type of notification the user will receive when a fax is transmitted. The value must be one of the valid NotifyType enumerated values.

### Table 7.11 NotifyType Enumerated Values

<table>
<thead>
<tr>
<th>Name</th>
<th>Numeric equivalent</th>
</tr>
</thead>
<tbody>
<tr>
<td>nNetworkBroadcast</td>
<td>0</td>
</tr>
<tr>
<td>nCustom1</td>
<td>1</td>
</tr>
<tr>
<td>nCustom2</td>
<td>2</td>
</tr>
<tr>
<td>nCustom3</td>
<td>3</td>
</tr>
<tr>
<td>nCustom4</td>
<td>4</td>
</tr>
<tr>
<td>nCustom5</td>
<td>5</td>
</tr>
<tr>
<td>nCustom6</td>
<td>6</td>
</tr>
<tr>
<td>nCustom7</td>
<td>7</td>
</tr>
<tr>
<td>nCustom8</td>
<td>8</td>
</tr>
<tr>
<td>nCustom9</td>
<td>9</td>
</tr>
<tr>
<td>nCCMail</td>
<td>10</td>
</tr>
<tr>
<td>nMSMail</td>
<td>11</td>
</tr>
<tr>
<td>nGroupWise</td>
<td>12</td>
</tr>
<tr>
<td>nTRS</td>
<td>13</td>
</tr>
<tr>
<td>nCX3</td>
<td>14</td>
</tr>
<tr>
<td>nPager</td>
<td>15</td>
</tr>
<tr>
<td>nNotes</td>
<td>16</td>
</tr>
<tr>
<td>nExchange</td>
<td>17</td>
</tr>
<tr>
<td>nSMTP</td>
<td>18</td>
</tr>
</tbody>
</table>
**NumberOfFaxesOwned**
Returns the total number of faxes in the Faxes Collection associated with this user. Use the Faxes property to retrieve the collection of faxes.

**Password**
Specifies the user’s password. Only the user and the administrator can access this property. See also the IsAdministrator property and the ChangePassword method.

**PersonalFaxNumber**
Specifies the fax number for this user.

**PersonalVoiceNumber**
Specifies the voice number for this user. See also the DIDNumber property.

**PhoneBook**
Returns the PhoneBook Collection for the user. See also the PhoneBook Collection.

**ReceivedAutoprintPrinter**
Specifies the printer where the received faxes will print if the IsAutoprintReceivedBodyEnabled, IsAutoprintReceivedCoversheetEnabled, or IsAutoprintReceivedHistoryEnabled property is set to True or 1.

**ReceiveNotify**
Specifies how the user wants to be notified of received faxes. The value must be one of the ReceiveNotifyType enumerated values. If the notify type is rnOnce or rnPeriodically, the NotifyType property must also be set.

<table>
<thead>
<tr>
<th>Name</th>
<th>Numeric equivalent</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>rnNone</td>
<td>0</td>
<td>No notification will be sent.</td>
</tr>
<tr>
<td>rnOnce</td>
<td>1</td>
<td>Notification will only be sent on initial receipt of the fax.</td>
</tr>
<tr>
<td>rnPeriodically</td>
<td>2</td>
<td>Notification will be sent periodically according to the specifications set by the administrator.</td>
</tr>
</tbody>
</table>

**ReceivedAutoprintPrinterID**
If autoprinting is enabled, this returns a string value for the printer ID to which received faxes will be printed.
Users, Groups, and Folders

**RouteFormat**
Specifies the format in which to send faxes that this user forwards using the ForwardToNewFaxNumber property in the Fax object. This converts the fax to the specified format such as a TIFF or PDF before sending to a recipient outside of the RightFax system. The format must be one of the valid UserRouteFormatType enumerated values. See also the RouteInfo and RouteType properties.

<table>
<thead>
<tr>
<th>Name</th>
<th>Numeric equivalent</th>
</tr>
</thead>
<tbody>
<tr>
<td>urfDCX</td>
<td>0</td>
</tr>
<tr>
<td>urfPCX</td>
<td>1</td>
</tr>
<tr>
<td>urfTiffGroup3</td>
<td>2</td>
</tr>
<tr>
<td>urfTiffGroup4</td>
<td>3</td>
</tr>
<tr>
<td>urfGIF</td>
<td>4</td>
</tr>
<tr>
<td>urfPDF</td>
<td>5</td>
</tr>
<tr>
<td>urfPDFWithThumbnails</td>
<td>6</td>
</tr>
<tr>
<td>urfCPC</td>
<td>7</td>
</tr>
<tr>
<td>urfRFX</td>
<td>8</td>
</tr>
<tr>
<td>urfPNG</td>
<td>9</td>
</tr>
</tbody>
</table>

**RouteInfo**
Specifies the information that the server requires for routing. This will vary for each route type. See the RightFax Administrator’s Guide fax server documentation for more information.

**RouteType**
Specifies the type of system to which routed faxes are being sent, such as CCMail, SAP, or Lotus Notes. The routing type must be one of the valid RouteType enumerated values. Depending on the recipient system, you may also need to specify the RouteInfo and RouteFormat properties.
Table 7.14  RouteType Enumerated Values

<table>
<thead>
<tr>
<th>Name</th>
<th>Numeric Equivalent</th>
</tr>
</thead>
<tbody>
<tr>
<td>rRightFax</td>
<td>0</td>
</tr>
<tr>
<td>rCCMail</td>
<td>1</td>
</tr>
<tr>
<td>rMSMail</td>
<td>2</td>
</tr>
<tr>
<td>rFile</td>
<td>4</td>
</tr>
<tr>
<td>rOCR</td>
<td>5</td>
</tr>
<tr>
<td>rGroupWiseMail</td>
<td>6</td>
</tr>
<tr>
<td>rNotes</td>
<td>7</td>
</tr>
<tr>
<td>rXRoute</td>
<td>8</td>
</tr>
<tr>
<td>rTRS</td>
<td>9</td>
</tr>
<tr>
<td>rCallXPress</td>
<td>10</td>
</tr>
<tr>
<td>rExchange</td>
<td>11</td>
</tr>
<tr>
<td>rSMTP</td>
<td>12</td>
</tr>
<tr>
<td>rANI</td>
<td>13</td>
</tr>
<tr>
<td>rSAP1</td>
<td>14</td>
</tr>
<tr>
<td>rSAP2</td>
<td>15</td>
</tr>
<tr>
<td>rSAP3</td>
<td>16</td>
</tr>
<tr>
<td>rSAP4</td>
<td>17</td>
</tr>
<tr>
<td>rSAP5</td>
<td>18</td>
</tr>
<tr>
<td>rSAP6</td>
<td>19</td>
</tr>
<tr>
<td>rSAP7</td>
<td>20</td>
</tr>
<tr>
<td>rSAP8</td>
<td>21</td>
</tr>
<tr>
<td>rSAP9</td>
<td>22</td>
</tr>
<tr>
<td>rSAP10</td>
<td>23</td>
</tr>
<tr>
<td>rSAP11</td>
<td>24</td>
</tr>
<tr>
<td>rSAP12</td>
<td>25</td>
</tr>
<tr>
<td>rSAP13</td>
<td>26</td>
</tr>
<tr>
<td>rSAP14</td>
<td>27</td>
</tr>
<tr>
<td>rSAP15</td>
<td>28</td>
</tr>
<tr>
<td>rSAP16</td>
<td>29</td>
</tr>
</tbody>
</table>
**SendNotifyOnIncompleteFirstTime**
This Boolean property specifies whether or not a single notification will be sent to the user if a sent fax does not contain enough information to complete transmission. The system uses the setting in the NotifyType property to determine the type of notification for the user. True or 1 indicates notification will be sent; False or 0 indicates it will not be sent.

**SendNotifyOnIncompletePeriodically**
This Boolean property specifies whether or not repeated notifications will be sent to the user if a sent fax does not contain enough information to complete transmission. The system uses the setting in the NotifyType property to determine the type of notification for the user. The number and frequency of the periodic notifications is determined by the administrator. True or 1 indicates notification will be sent; False or 0 indicates it will not be sent.

**SendNotifyOnNoHoldForPreview**
This Boolean property specifies whether or not notification will be sent to the user when a sent fax is being transmitted without a preview. The system uses the setting in the NotifyType property to determine the type of notification for the user. True or 1 indicates notification will be sent; False or 0 indicates it will not be sent.

**SendNotifyOnSendFailedWillRetry**
This Boolean property specifies whether or not notification will be sent to the user when a sent fax fails transmission, even if the system will attempt to resend the fax. The system uses the setting in the NotifyType property to determine the type of notification for the user. True or 1 indicates notification will be sent; False or 0 indicates it will not be sent.

**SendNotifyOnSendFailure**
This Boolean property specifies whether or not notification will be sent to the user when a sent fax fails transmission. The system uses the setting in the NotifyType property to determine the type of notification for the user. True or 1 indicates notification will be sent; False or 0 indicates it will not be sent.

**SendNotifyOnSendHeldForApproval**
This Boolean property specifies whether or not notification will be sent to the user when a sent fax is held waiting for approval. The system uses the setting in the NotifyType property to determine the type of notification for the user. True or 1 indicates notification will be sent; False or 0 indicates it will not be sent.

**SendNotifyOnSendingFirstTime**
This Boolean property specifies whether or not a single notification will be sent to the user when a sent fax begins transmission. The system uses the setting in the NotifyType property to determine the type of notification for the user. True or 1 indicates notification will be sent; False or 0 indicates it will not be sent.
**SendNotifyOnSendingPeriodically**
This Boolean property specifies whether or not repeated notifications will be sent to the user as a sent fax is transmitted. The system uses the setting in the NotifyType property to determine the type of notification for the user. The number and frequency of the periodic notifications is determined by the administrator. True or 1 indicates notification will be sent; False or 0 indicates it will not be sent.

**SendNotifyOnSentSuccessfully**
This Boolean property specifies whether or not notification will be sent to the user when a sent fax completes successful transmission. The system uses the setting in the NotifyType property to determine the type of notification for the user. True or 1 indicates notification will be sent; False or 0 indicates it will not be sent.

**SentAutoprintPrinter**
Specifies the Printer object where the sent faxes will print if the IsAutoprintSentEnabled property is set to True or 1.

**SentAutoprintPrinterID**
Specifies a string value for the Printer ID where the sent faxes will print if the IsAutoprintSentEnabled property is set to True or 1.

**UpdateInterval**
Specifies the time in minutes between automatic updates of the user’s fax list.

**UserName**
Specifies the full name of the user.

**VoiceMailSubscriberID**
Specifies the voice mail subscriber ID associated with the user, if any.

**WebClientImageFormat**
Specifies the format to which images will be converted for distribution over the Web Client. The format must be one of the valid UserRouteFormatType values.
Users, Groups, and Folders

Table 7.15 UserRouteFormatType Enumerated Values

<table>
<thead>
<tr>
<th>Name</th>
<th>Numeric equivalent</th>
</tr>
</thead>
<tbody>
<tr>
<td>urfDCX</td>
<td>0</td>
</tr>
<tr>
<td>urfPCX</td>
<td>1</td>
</tr>
<tr>
<td>urfTiffGroup3</td>
<td>2</td>
</tr>
<tr>
<td>urfTiffGroup4</td>
<td>3</td>
</tr>
<tr>
<td>urfGIF</td>
<td>4</td>
</tr>
<tr>
<td>urfPDF</td>
<td>5</td>
</tr>
<tr>
<td>urfPDFWithThumbnails</td>
<td>6</td>
</tr>
<tr>
<td>urfCPC</td>
<td>7</td>
</tr>
<tr>
<td>urfRFX</td>
<td>8</td>
</tr>
<tr>
<td>urfPNG</td>
<td>9</td>
</tr>
</tbody>
</table>

WillReceivedFaxesHaveBillingCodesAssociated
This Boolean property specifies whether or not the user’s default billing codes will automatically be associated with incoming faxes. True or 1 indicates the billing codes will be automatically assigned; False or 0 indicates they will not. See also the DefaultBillingCode property.

Methods

ChangePassword
Changes the password for this user by specifying a new password. Only this user or the administrator can change the password.

Delete
Removes the object from the server.

Refresh
Re-loads the list of user objects from the fax server.

Save
Writes the object to the server.
Example

The following example lists all users in a list box called lstRFUser.

```
For Each User In MyFaxServer.Users
    lstRFUser.AddItem User.ID
Next
```

Users Collection

A Users collection is a set of one or more User objects.

Properties

- **Count**
  Returns the number of objects in the collection.

- **Create**
  Returns a new object.

- **Item**
  Returns the object with the specified item number. Each object in the collection is numbered, starting with 1.

Methods

- **Add**
  Adds an object to the collection

- **Refresh**
  Re-loads the list of form objects from the fax server.

- **Remove**
  Removes a specified object from the collection but not from the server. To remove an object from the server, use the Delete method in the object you want to delete.

- **RemoveAll**
  Removes all objects from the collection but not from the server. To remove an object from the server, use the Delete method in the object you want to delete.
Exercise 1

In this exercise you will create a group object and display all groups in a list box. You will also create a user object and place it into a group that you select from the list box.

1. Add the ability to create a group on your RightFax Server.
   - Create a text box where you can add the name of a group.
   - Add a button that allows you to create the group object based upon the name specified in the text box.
   - Create a list box which displays all the groups available on the RightFax server.

2. Add the ability to create a new user on your RightFax Server.
   - Create a text box(es) to input the id property (and any other properties you wish to set) that will be associated with the User object.
   - Add a button that allows you to create the user object based upon the information specified in the text boxes.
Section 8
Phonebooks and Printers

Phonebooks and Printers

Upon completion of this section, you should be able to:

- Utilize the phonebook objects.
- Describe the properties, methods and events associated with Phonebooks.
- Describe the properties, methods and events associated with Printers.
Phonebook and Printer Objects Overview

A phonebook in the RightFax system is defined by each user and accessed when sending a fax. Like other types of phonebooks, it contains the information about the recipients to whom the user regularly sends faxes.

The printer objects are also defined by the user and accessed by the Fax objects. Within the User object, there are properties that specify when a fax is automatically printed and which printers are default printers for each type of print job.

PhoneBook Collection

A PhoneBook is a set of one or more PhoneItem objects. The PhoneBook Collection is called by the user with the PhoneBook property in the User object.

Properties

- **Count**
  Returns the number of objects in the collection.

- **Create**
  Returns a new object.

- **Item**
  Returns the object with the specified item number. Each object in the collection is numbered, starting with 1.

Methods

- **Add**
  Adds an object to the collection.

- **Refresh**
  Re-loads the list of phonebook objects from the fax server.

- **Remove**
  Removes a specified object from the collection but not from the server. To remove an object from the server, use the Delete method in the object you want to delete.

- **RemoveAll**
  Removes all objects from the collection but not from the server. To remove an object from the server, use the Delete method in the object you want to delete.
PhoneItem

There are two types of PhoneItem objects, PhoneItemElement and PhoneItemGroup objects. Both have the same basic set of properties and methods, described here under PhoneItem. In addition, PhoneItemElement objects contain information about a specific recipient, such as address and fax numbers. A PhoneItemGroup is actually a collection of PhoneItemElements and may also contain other PhoneItemGroup collections.

Properties

**Handle**
Returns the code used internally by the RightFax system to identify each unique object.

**ID**
Specifies an ID you can assign to identify each unique object. If you specify a value without specifying a property name, the system will assign that value as the object ID.

**IsCertifiedDeliveryEnabled**
This Boolean property specifies whether or not documents sent to this PhoneItem should be sent via certified delivery.

**IsEmailRecipient**
This Boolean property specifies whether or not the phone item is an e-mail recipient. True or 1 indicates the phone item is an e-mail recipient; False or 0 indicates it is a fax recipient.

**IsExternallyPublished**
This Boolean property specifies whether or not the phone item is available to applications other than the RightFax software. True or 1 indicates the phone item is available; False or 0 indicates it is not.

**IsHiddenFromCCLList**
This Boolean property specifies whether or not the phone item will appear on the cover sheet in the cc list. True or 1 indicates the phone item will not appear; False or 0 indicates it will.

**IsPublished**
This Boolean property specifies whether or not the phone item is available for viewing by other users of the RightFax software. True or 1 indicates the phone item is available; False or 0 indicates it is not.
IsReadOnly
This Boolean property specifies whether or not the phone item can be modified by users of the RightFax software other than the owner. True or 1 indicates the phone item can not be modified; False or 0 indicates it can.

Owner
Specifies the user who is the owner of the phone item.

TypeOfPhoneItem
Specifies the type of phone item. The specified value must be one of the valid PhoneItemType enumerated values.

Table 8.1  PhoneItemType Enumerated Values

<table>
<thead>
<tr>
<th>Name</th>
<th>Numeric equivalent</th>
</tr>
</thead>
<tbody>
<tr>
<td>piElement</td>
<td>0</td>
</tr>
<tr>
<td>piGroup</td>
<td>1</td>
</tr>
</tbody>
</table>

Methods

Delete
Removes the object from the server.

Save
Writes the object to the server.
PhonItemElement

Each phone item element contains the addressing information for a single recipient. The PhonItemElement also contains the information specified in the PhonItem to which the element belongs, including all properties and methods.

Properties

**Address**
Specifies the recipient’s street address.

**BillingCode**
Specifies the billing code that will be assigned to all faxes sent to this recipient. This may be overridden by the BillingCode property in the Fax object.

**CityState**
Specifies the recipient’s city and state.

**Company**
Specifies the recipient’s company name.

**FaxNumberAlternate**
Specifies the alternate fax number for the recipient.

**FaxNumberPrimary**
Specifies the primary fax number for the recipient.

**Name**
Specifies the recipient’s full name.

**Notes**
Specifies notes associated with the recipient.

**SecureCSID**
Specifies the secure caller subscriber identification of the recipient’s fax machine, if any.

**VoiceNumberAlternate**
Specifies the recipient’s alternate voice telephone number.

**VoiceNumberPrimary**
Specifies the recipient’s voice telephone number.
Example

The following example creates a phonebook entry for a user named Joe.

```
Dim obUser As RFCOMAPILib.User
Dim obPhone as RFCOMAPILib.PhoneItemElement
Public Sub AddPhoneItem()
    Set obUser = MyFaxServer.User ("Joe")
    Set obPhone = MyFaxServer.CreateObject (coPhoneItemElement)
    obPhone.Name = "To name"
    obPhone.Owner = obUser
    obPhone.Save
End Sub
```

PhoneItemGroup Collection

A PhoneItemGroup is a set of one or more PhoneItem objects. The PhoneItemGroup also contains the information specified in the PhoneItem to which the group belongs, including all properties and methods. Each PhoneItemGroup may also contain other PhoneItemGroup collections.

Properties

- **Count**
  Returns the number of objects in the collection.

- **Item**
  Returns the object with the specified item number. Each object in the collection is numbered, starting with 1.

Methods

- **Add**
  Adds an object to the collection

- **Load**
  Loads the elements into the collection. This prevents recursive loading of the nested phonebook groups.

- **Remove**
  Removes a specified object from the collection but not from the server. To remove an object from the server, use the Delete method in the object you want to delete.
RemoveAll
Removes all objects from the collection but not from the server. To remove an object from the server, use the Delete method in the object you want to delete.

Printer
The Printer object defines a printer for the RightFax system.

Properties

DefaultPrintSize
Specifies the size of the paper in the default tray of the printer. The default tray is specified by the DefaultPrintSource property. The paper size must be one of the valid PrintSizeType enumerated values.

Table 8.2 PrintSizeType Enumerated Values

<table>
<thead>
<tr>
<th>Name</th>
<th>Numeric equivalent</th>
</tr>
</thead>
<tbody>
<tr>
<td>psLetter</td>
<td>1</td>
</tr>
<tr>
<td>psLegal</td>
<td>2</td>
</tr>
<tr>
<td>psA4</td>
<td>3</td>
</tr>
<tr>
<td>psLetterLegal</td>
<td>4</td>
</tr>
<tr>
<td>psA4Legal</td>
<td>5</td>
</tr>
</tbody>
</table>

DefaultPrintSource
Specifies the default location of the paper in the printer. The size of paper in this default location is specified by the DefaultPrintSize property. The paper source must be one of the valid PrintPaperSourceType values.
Phonebooks and Printers

Table 8.3 PrintPaperSourceType Enumerated Values

<table>
<thead>
<tr>
<th>Name</th>
<th>Numeric equivalent</th>
</tr>
</thead>
<tbody>
<tr>
<td>ppsUpperOrTray1</td>
<td>1</td>
</tr>
<tr>
<td>ppsLowerOrTray2</td>
<td>2</td>
</tr>
<tr>
<td>ppsManual</td>
<td>3</td>
</tr>
<tr>
<td>ppsTray3</td>
<td>4</td>
</tr>
<tr>
<td>ppsMPTrayOrTray4</td>
<td>5</td>
</tr>
<tr>
<td>ppsHighCapacityOrTray5</td>
<td>6</td>
</tr>
</tbody>
</table>

**Description**
Specifies a description for this printer.

**Handle**
Returns the database handle used internally by the RightFax system to identify each unique object.

**ID**
Specifies an ID you can assign to identify each unique object. If you specify a value without specifying a property name, the system will assign that value as the object ID.

**PrintDirectlyToAnIPAddress**
This Boolean property specifies whether or not the printer is printing to an IP address. True or 1 indicates the printer is printing to an IP address; False or 0 indicates it is not.

**PrinterType**
Specifies the brand and model of printer. The printer type must be one of the valid PrinterType enumerated values.
### Table 8.4 PrinterType Enumerated Values

<table>
<thead>
<tr>
<th>Name</th>
<th>Numeric equivalent</th>
</tr>
</thead>
<tbody>
<tr>
<td>pNone</td>
<td>0</td>
</tr>
<tr>
<td>pLaserJet</td>
<td>1</td>
</tr>
<tr>
<td>pLaserJetII</td>
<td>2</td>
</tr>
<tr>
<td>pDeskJet</td>
<td>3</td>
</tr>
<tr>
<td>pLaserJetIII</td>
<td>4</td>
</tr>
<tr>
<td>pDeskJet500</td>
<td>5</td>
</tr>
<tr>
<td>pPostScriptLevel1</td>
<td>6</td>
</tr>
<tr>
<td>pLaserJet4</td>
<td>7</td>
</tr>
<tr>
<td>pPPDS</td>
<td>8</td>
</tr>
<tr>
<td>pXIP</td>
<td>9</td>
</tr>
<tr>
<td>pXeroxDCS20</td>
<td>10</td>
</tr>
<tr>
<td>pXeroxDCS35</td>
<td>11</td>
</tr>
<tr>
<td>pLaserJet3si</td>
<td>12</td>
</tr>
<tr>
<td>pLaserJet4si</td>
<td>13</td>
</tr>
<tr>
<td>pLaserJet5si</td>
<td>14</td>
</tr>
<tr>
<td>pATIImagePrinter</td>
<td>15</td>
</tr>
<tr>
<td>pLaserJet5siMopier</td>
<td>16</td>
</tr>
<tr>
<td>pXeroxN24</td>
<td>17</td>
</tr>
<tr>
<td>pXeroxDCS220</td>
<td>18</td>
</tr>
<tr>
<td>pXeroxDCS230</td>
<td>19</td>
</tr>
<tr>
<td>pXeroxDCS240</td>
<td>20</td>
</tr>
<tr>
<td>pXeroxDCS255</td>
<td>21</td>
</tr>
<tr>
<td>pXeroxDCS265</td>
<td>22</td>
</tr>
<tr>
<td>pXeroxRicoh250</td>
<td>23</td>
</tr>
<tr>
<td>pXeroxLaserJet8000</td>
<td>24</td>
</tr>
<tr>
<td>pXeroxGDI</td>
<td>25</td>
</tr>
<tr>
<td>pXeroxDCS460</td>
<td>26</td>
</tr>
<tr>
<td>pXeroxDCS470</td>
<td>27</td>
</tr>
<tr>
<td>pXeroxDCS480</td>
<td>28</td>
</tr>
</tbody>
</table>
QueueName
Specifies the name of the queue on the server. The server is specified by the ServerName property.

Request
Generates a PrintRequest object, which the server uses to print the fax.

ServerName
Specifies the name of the server hosting the print queue specified in the QueueName property.

Methods

Delete
Removes the print record from the server.

Save
Writes the print record to the server.

Printers Collection

A Printers Collection is a set of one or more Printer objects.

Properties

Count
Returns the number of objects in the collection.

Create
Creates a new object.

Item
Returns the object with the specified item number. Each object in the collection is numbered, starting with 1.

Methods

Add
Adds an existing printer object to the collection

Refresh
Re-loads the list of printer objects from the fax server.
Remove
Removes a specified object from the collection but not from the server. To remove an object from the server, use the Delete method in the object you want to delete.

RemoveAll
Removes all objects from the collection but not from the server. To remove an object from the server, use the Delete method in the object you want to delete.

PrintRequest
A PrintRequest contains the specific information needed to print a Fax object, such as resolution and paper size information. Use the Request method in the Printer object to create a print request. Use the print request to print the fax.

Properties

**CollateOutput**
Specifies whether or not multiple copies of multi-page faxes will be collated, producing one copy of the entire fax before beginning the second copy. This property only applies to printer types that support this feature. True or 1 indicates the output will be collated; False or 0 indicates it will not be collated.

**Duplex**
Specifies how the printer should create two-sided copies, if the printer supports this feature. The duplex type must be one of the valid PrintDuplexType enumerated values.

<table>
<thead>
<tr>
<th>Name</th>
<th>Numeric equivalent</th>
</tr>
</thead>
<tbody>
<tr>
<td>pdDefault</td>
<td>0</td>
</tr>
<tr>
<td>pdSingle</td>
<td>1</td>
</tr>
<tr>
<td>pdLongEdge</td>
<td>2</td>
</tr>
<tr>
<td>pdShortEdge</td>
<td>3</td>
</tr>
</tbody>
</table>

**EndPage**
Specifies the last page of a multi-page output that will be printed. To print all pages, specify -1 as the end page.
**NumberOfCopies**
Specifies the number of copies to print.

**OutputBin**
Specifies where in the printer the printed output will be sent. For example, if there is a separate mechanism for stapling, the printed pages may go to the poBinFinisher bin. The bin must be one of the valid PrintOutputBinType values.

<table>
<thead>
<tr>
<th>Name</th>
<th>Numeric equivalent</th>
</tr>
</thead>
<tbody>
<tr>
<td>poBinDefault</td>
<td>0</td>
</tr>
<tr>
<td>poBinUpper</td>
<td>1</td>
</tr>
<tr>
<td>poBinLower</td>
<td>2</td>
</tr>
<tr>
<td>poBinFinisher</td>
<td>3</td>
</tr>
</tbody>
</table>

**PaperSource**
Specifies which printer tray contains the paper to be used for this output. The paper source must be one of the valid PrintPaperSourceType values. The default value is set for the printer with the DefaultPrintSource property in the Printer object.

<table>
<thead>
<tr>
<th>Name</th>
<th>Numeric equivalent</th>
</tr>
</thead>
<tbody>
<tr>
<td>ppsDefault</td>
<td>0</td>
</tr>
<tr>
<td>ppsUpperOrTray1</td>
<td>1</td>
</tr>
<tr>
<td>ppsLowerOrTray2</td>
<td>2</td>
</tr>
<tr>
<td>ppsManual</td>
<td>3</td>
</tr>
<tr>
<td>ppsTray3</td>
<td>4</td>
</tr>
<tr>
<td>ppsMPTrayOrTray4</td>
<td>5</td>
</tr>
<tr>
<td>ppsHighCapacityOrTray5</td>
<td>6</td>
</tr>
</tbody>
</table>
Phonebooks and Printers

PrintCover
This Boolean property specifies whether or not to print the cover sheet for sent faxes. True or 1 indicates the cover sheet will print; False or 0 indicates it will not.

PrinterID
Specifies an ID you can assign to identify the printer. If you specify a value without specifying a property name, the system will assign that value as the printer ID.

PrintTiffHeader
This Boolean property specifies whether or not the TIFF image header will print. True or 1 indicates it will print; False or 0 indicates it will not.

PrintTransmissionHistory
This Boolean property specifies whether or not to print the transmission history for the fax. True or 1 indicates the history will print; False or 0 indicates it will not.

Resolution
Specifies the print resolution for the fax. Note that this is not the same as the Fax object property IsFineMode, which specifies the resolution at which the fax image is stored. The print resolution must be one of the valid PrintResolutionType values, each of which will vary with the type of output printer.

Table 8.8 PrintResolutionType Enumerated Values

<table>
<thead>
<tr>
<th>Name</th>
<th>Numeric equivalent</th>
</tr>
</thead>
<tbody>
<tr>
<td>prHigh</td>
<td>0</td>
</tr>
<tr>
<td>prMedium</td>
<td>1</td>
</tr>
<tr>
<td>prLow</td>
<td>2</td>
</tr>
</tbody>
</table>

Size
Specifies the size of the paper on which the fax will print. The paper size must be one of the valid PrintSizeType values. The default value for each printer is specified with the DefaultPrintSize property.
Phonebooks and Printers

Table 8.9  PrintSizeType Enumerated Values

<table>
<thead>
<tr>
<th>Name</th>
<th>Numeric equivalent</th>
</tr>
</thead>
<tbody>
<tr>
<td>psDefault</td>
<td>0</td>
</tr>
<tr>
<td>psLetter</td>
<td>1</td>
</tr>
<tr>
<td>psLegal</td>
<td>2</td>
</tr>
<tr>
<td>psA4</td>
<td>3</td>
</tr>
<tr>
<td>psLetterLegal</td>
<td>4</td>
</tr>
<tr>
<td>psA4Legal</td>
<td>5</td>
</tr>
</tbody>
</table>

**StapleOutput**
This Boolean property specifies whether or not multi-page output should be stapled. This only applies to printers that support this feature. True or 1 indicates the output will be stapled; False or 0 indicates it will not be stapled.

**StartPage**
Specifies the first page that will be printed.

**Methods**
None.
Section 9
Fax History Objects

Objectives

Upon completion of this section, you should be able to:

- Write an application that utilizes Fax History Objects.
- Specify what fax histories are available.
- Specify what properties can be interrogated for each of the history objects.
Fax History Object Overview

A history object is created when activity occurs for a fax, such as transmission, printing, or conversion to OCR. The base class of all fax histories is the FaxHistory object. There are nine types of FaxHistory objects, each of which contains the core FaxHistory object properties as well as unique properties for that type. For example, when a fax is approved, a core FaxHistory object is created with a history type of FaxHistoryApproval. This indicates that the object will also contain the properties specified by the ApprovalFaxHistory object, such as the name of the user who approved the fax. If the type is FaxHistoryPrint, the FaxHistory object will contain information about the printer and number of pages printed, as specified in the PrintFaxHistory object.

FaxHistories Collection

A FaxHistories Collection is a set of one or more FaxHistory objects.

Properties

Count
Returns the number of objects in the collection.

Item
Returns the object with the specified item number. Each object in the collection is numbered, starting with 1.

Methods

None.

FaxHistory

A FaxHistory object is created when activity occurs for a fax, such as transmission, printing, or conversion to OCR. There are nine types of FaxHistory objects, each of which contains the core FaxHistory object properties as well as unique properties for that type. For example, if the type is FaxHistoryApproval, then the FaxHistory object is actually an ApprovalFaxHistory object and will contain the name of the user who approved the fax. If the type is FaxHistoryPrint, the FaxHistory object is actually a PrintFaxHistory object and will contain information about the printer and number of pages printed.
Properties

**DateTime**
Specifies the date and time the fax history object was created.

**Handle**
Returns the database handle used internally by the RightFax system to identify each unique object.

**TypeOfHistory**
Specifies the type of fax history object. This object will contain additional properties based on the fax history type. The history type must be one of the FaxHistoryType enumerated values.

### Table 9.1 FaxHistoryType Enumerated Values

<table>
<thead>
<tr>
<th>Name</th>
<th>Numeric equivalent</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>FaxHistoryTransmission</td>
<td>0</td>
<td>See the TransmissionFaxHistory object.</td>
</tr>
<tr>
<td>FaxHistoryRoute</td>
<td>1</td>
<td>See the RouteFaxHistory object.</td>
</tr>
<tr>
<td>FaxHistoryPrint</td>
<td>2</td>
<td>See the PrintFaxHistory object.</td>
</tr>
<tr>
<td>FaxHistoryConversionError</td>
<td>3</td>
<td>See the ConversionErrorFaxHistory object.</td>
</tr>
<tr>
<td>FaxHistoryOCR</td>
<td>4</td>
<td>See the OCRFaxHistory object.</td>
</tr>
<tr>
<td>FaxHistoryNetForward</td>
<td>5</td>
<td>See the NetForwardFaxHistory object.</td>
</tr>
<tr>
<td>FaxHistoryApproval</td>
<td>6</td>
<td>See the ApprovalFaxHistory object.</td>
</tr>
<tr>
<td>FaxHistoryDisapproval</td>
<td>7</td>
<td>See the DisapprovalFaxHistory object.</td>
</tr>
<tr>
<td>FaxHistoryFileRoute</td>
<td>8</td>
<td>See the FileRouteFaxHistory object.</td>
</tr>
<tr>
<td>FaxHistorySecureDoc</td>
<td>9</td>
<td>See the SecureDocFaxHistory object.</td>
</tr>
</tbody>
</table>
Fax History Objects

Methods

None.

Example

The following example retrieves the transmission history for a user’s faxes.

Dim obFaxHistory As RFCOMAPILib.FaxHistory
Dim obFax As RFCOMAPILib.Fax
For Each obFax In MyFaxServer.Faxes ("UserID")
    For Each obFaxHistory In obFax.Histories
        If obFaxHistory.TypeOfHistory = FaxHistoryTransmission Then
            Dim obFaxTransHist As RFCOMAPILib.TransmissionFaxHistory
            Set obFaxTransHist = obFaxHistory
            sDesc = obFaxTransHist.Description
            MsgBox sDesc
        End If
    Next
Next

ApprovalFaxHistory

This type of history object is created when a fax has been approved. The base FaxHistory object will have a history type of FaxHistoryApproval and will also contain the following properties.

Properties

Approver
Returns the user who approved the fax.

Notes
Returns the notes, if any, that the user entered when approving the fax.

ConversionErrorFaxHistory

This type of history object is created when conversion of a fax is attempted, either to or from an image file. The base FaxHistory object will have a history type of FaxHistoryConversionError and will also contain the following properties.
Properties

**ConversionType**
Returns the type of conversion that caused the failure.

**Error1**
Returns the first field of the error message generated by the conversion process.

**Error2**
Returns the second field of the error message generated by the conversion process.

**ErrorMessage**
Returns the name or ID of the error message generated by the conversion process.

**DisapprovalFaxHistory**

This type of history object is created when a fax is disapproved. The base FaxHistory object will have a history type of FaxHistoryDisapproval and will also contain the following properties.

Properties

**Disapprover**
Returns the user who disapproved the fax.

**Notes**
Returns the notes, if any, the user entered when disapproving the fax.

**FileRouteFaxHistory**

This type of history object is created when a fax is routed to a network location. The base FaxHistory object will have a history type of FaxHistoryRoute and will also contain the following properties.

Properties

**Error**
Returns the error message, if any, generated by the system to which the routing was attempted.
Fax History Objects

**RoutePath**
Returns the full network path to which the fax was routed or to which routing was attempted.

**NetForwardFaxHistory**
This type of history object is created when a fax is forwarded to users within the RightFax system. The base FaxHistory object will have a history type of FaxHistoryNetForward and will also contain the following properties.

**Properties**

- **Notes**
  Returns the notes, if any, the user entered when requesting that the fax be forwarded.

- **PreviousOwner**
  Returns the user who was the owner of the fax before it was forwarded.

- **UsersForwardedTo**
  Returns the Users collection that contains the users to whom the fax was forwarded.

**OCRFaxHistory**
This type of history object is created when a fax is converted from image file to text. The base FaxHistory object will have a history type of FaxHistoryOCR and will also contain the following properties.

**Properties**

- **Error1**
  Returns the first field of the error message, if any, generated by the OCR conversion process.

- **Message**
  Returns a status message generated by the OCR conversion process.

- **TimeToOCR**
  Returns the time, in seconds, that it took to convert the fax image to text.
Fax History Objects

**PrintFaxHistory**

This type of history object is created when a fax is sent to the printer. The base FaxHistory object will have a history type of FaxHistoryPrint and will also contain the following properties.

**Properties**

- **CopiesPrinted**
  Returns the number of copies of the fax that were printed.

- **Error1**
  Returns the first field of the error message, if any, generated by the print process.

- **Error2**
  Returns the second field of the error message, if any, generated by the print process.

- **Message**
  Returns a status message generated by the print process.

- **PagesPrinted**
  Returns the total number of pages printed.

- **Printer**
  Returns the printer to which the print job was sent.

- **TimeToPrint**
  Returns the time, in seconds, it took to print the fax.

**RouteFaxHistory**

This type of history object is created when a fax was routed to other recipients who are not on the RightFax system. The base FaxHistory object will have a history type of FaxHistoryRoute and will also contain the following properties.

**Properties**

- **IsAutomaticallyRouted**
  This Boolean property returns True or 1 if the fax was automatically routed, False if it was manually routed.

- **Notes**
  Returns notes entered by the user when submitting the routing request.
Fax History Objects

**PreviousOwner**
Returns the user who was the owner of the fax before it was routed.

**SecureDocHistory**
This type of history object is created when a fax document is sent to an e-mail address. The base FaxHistory object will have a history type of FaxHistorySecureDoc and will also contain the following properties.

**Properties**

**CurrentStatus**
Returns a short description of the status of the document when the history element was created.

**RemoteIP**
Returns the IP address of the web client that downloaded the certified document.

**SMTPHost**
Returns the SMTP mail server that the e-mail was delivered to for forwarding.

**SMTPSender**
Returns the e-mail address of the user who sent the fax.

**TerminationStatus**
Returns the transmission status when the transmission is complete. This will be one of these values:

- TERMSTAT_SUCCESSRECV
- TERMSTAT_TXRXERROR
- TERMSTAT_UNKNOWN

**UserAgent**
Returns the type of web browser that was used to download the certified document.

**TransmissionFaxHistory**
This type of history object is created when a fax is sent or received, even if the transmission is unsuccessful. The base FaxHistory object will have a history type of FaxHistoryTransmission and will also contain the following properties.
Fax History Objects

Properties

ANI
Returns the automatic number identification of the recipient’s fax, if available.

AOCDATA
Returns the array of change information, if any, about the modem over which the fax was transmitted.

BadPageCount
Returns the number of pages that were not transmitted successfully.

BoardType
Returns the type of fax board that was used to transmit the fax. This will be one of the valid TransmissionFaxHistoryBoardType enumerated values.

<table>
<thead>
<tr>
<th>Name</th>
<th>Numeric equivalent</th>
</tr>
</thead>
<tbody>
<tr>
<td>tfhbtBrooktrout</td>
<td>1</td>
</tr>
<tr>
<td>tfhbtGammalink</td>
<td>2</td>
</tr>
<tr>
<td>tfhbtBrooktroutNew</td>
<td>3</td>
</tr>
</tbody>
</table>

BrooktroutCallResultLineStatus
Returns the status of the line for a call. This is only valid if the board used is a Brooktrout board and the board type is tfhbtBrooktroutNew. See the Brooktrout documentation for more details.

BrooktroutCallResultStatus
Returns the status of a call. This is only valid if the board used is a Brooktrout board and the board type is tfhbtBrooktroutNew. See the Brooktrout documentation for more details.

BrooktroutFaxResultLineStatus
Returns the status of the fax line. This is only valid if the board used is a Brooktrout board and the board type is tfhbtBrooktroutNew. See the Brooktrout documentation for more details.

BrooktroutFaxResultStatus
Returns the status of the fax. This is only valid if the board used is a Brooktrout board and the board type is tfhbtBrooktroutNew. See the Brooktrout documentation for more details.
Fax History Objects

ChannelUsed
Returns the specific channel used to send or receive the fax.

ElapsedTime
Returns the time it took in seconds for the transmission.

FaxNumberSentTo
Returns the fax number that the fax was sent to.

GammalinkError
Returns an error generated by the Gammalink fax board. This is only valid if the board used is a Gammalink board and the board type is tfhbtGammalink. See the Gammalink documentation for more details.

GoodPageCount
Returns the number of pages successfully sent or received.

HangupStatus
Returns the status received from the fax board at the end of the transmission.

IsANIValid
This Boolean property returns True or 1 if the automatic number identification value is valid, False or 0 if it is not valid. See also the ANI property.

IsAOCValid
This Boolean property returns True or 1 if the array of change information is valid, False or 0 if it is not valid. See also the AOCData property.

ISDN Cause Value
Returns the value representing the cause of ISDN messages, if any.

IsFaxPartiallyRetried
This Boolean property returns True or 1 if the fax was unsuccessfully retried after a transmission error. It returns False or 0 if no retry took place.

IsRemotelySent
This Boolean property returns True or 1 if the fax was sent by a remote board server and returns False or 0 if it was sent by the local board server.

RemoteID
Returns the ID of the remote fax that sent the fax if the IsRemotelySent property is set to True or 1.

RemoteServer
Returns the remote server that sent the fax if the IsRemotelySent property is set to True or 1.
**TerminationStatus**
Returns the status of the line when the transmission is complete. This will be one of the enumerated values from the following table.

**Table 9.3 TransmissionFaxHistoryTerminationStatusType Enumerated Values**

<table>
<thead>
<tr>
<th>Name</th>
<th>Numeric equivalent</th>
</tr>
</thead>
<tbody>
<tr>
<td>tfhtstSuccessReceive</td>
<td>0x0</td>
</tr>
<tr>
<td>tfhtstBusy1</td>
<td>0x1</td>
</tr>
<tr>
<td>tfhtstBusy2</td>
<td>0x2</td>
</tr>
<tr>
<td>tfhtstReorderBusy</td>
<td>0x3</td>
</tr>
<tr>
<td>tfhtstRecall</td>
<td>0x4</td>
</tr>
<tr>
<td>tfhtstConfirm</td>
<td>0x5</td>
</tr>
<tr>
<td>tfhtstBlocked</td>
<td>0x6</td>
</tr>
<tr>
<td>tfhtstRing1</td>
<td>0x8</td>
</tr>
<tr>
<td>tfhtstRing2</td>
<td>0x9</td>
</tr>
<tr>
<td>tfhtstHuman</td>
<td>0x10</td>
</tr>
<tr>
<td>tfhtstAnswer</td>
<td>0x11</td>
</tr>
<tr>
<td>tfhtstDialtone</td>
<td>0x12</td>
</tr>
<tr>
<td>tfhtstSITDetect</td>
<td>0x13</td>
</tr>
<tr>
<td>tfhtstSITNOCIR</td>
<td>0x14</td>
</tr>
<tr>
<td>tfhtstTransmissionError</td>
<td>0x15</td>
</tr>
<tr>
<td>tfhtstHangNoLoopCurrent</td>
<td>0x16</td>
</tr>
<tr>
<td>tfhtstCallColide</td>
<td>0x17</td>
</tr>
<tr>
<td>tfhtstDialNoLoopCurrent</td>
<td>0x18</td>
</tr>
<tr>
<td>tfhtstRingNoAnswer</td>
<td>0x19</td>
</tr>
<tr>
<td>tfhtstGroup2Detected</td>
<td>0x1a</td>
</tr>
<tr>
<td>tfhtstQuiet</td>
<td>0x1c</td>
</tr>
<tr>
<td>tfhtstFaxMachine</td>
<td>0x20</td>
</tr>
<tr>
<td>tfhtstLocalInUse</td>
<td>0x21</td>
</tr>
<tr>
<td>tfhtstSilence</td>
<td>0x22</td>
</tr>
<tr>
<td>tfhtstUnknown</td>
<td>0x23</td>
</tr>
</tbody>
</table>
Exercise 1

In this exercise you will retrieve information about a forwarded fax.

1. Create a list box which will display the ids of all the users on your RightFax Server.

2. Create a NetForwardFaxHistory object that is associated with the user selected in the list box.

3. Add a button that will interrogate the PreviousOwner property of the NetForwardFaxHistory object and display the results in a list box on a separate form.
Section 10
Event Handling and Event Objects

Objectives

Upon completion of this section, you should be able to:

- Create an application that uses the event handler.
- Describe the different events associated with the FaxServer object.
- Describe the object returned by the event handler.
Using the Event Handler

The RightFax COM API can be set to perform tasks based on events such as when a fax is has completed sending or when a notification message is generated. In order to utilize the event handler within your application, you must complete the following steps.

1. Declare and set the object variables that will handle the event during the initialization of the Form.
2. Set the corresponding variable in the Event object that enables the event. For example, to make use of the OnArchiveEvent, you would have to set the WatchArchiveEvents property to True in the Event object.
3. Declare one or more functions that will be called by the COM API when the event happens.

Creating a Sample Project Using the Event Handler

In the following example, a project will be created that generates a message whenever a fax has completed sending.

1. In order to utilize the event handler within your application you must first declare the FaxServer and Events objects globally.

   ```vba
   Dim WithEvents MyFaxServerEventHandler As RFCOMAPILib.FaxServer
   Dim EventsObj As RFCOMAPILib.Events
   ```

2. Set the two object variables to handle events. Add the following code to the Form_Load method of frmMain in your project.

   ```vba
   Set MyFaxServerEventHandler = MyFaxServer
   Set EventsObj = MyFaxServer.Events
   ```

3. For this example, you must set the GenerateCompletionEvent property equal to true for the fax object you create. You should add this code before calling the Send method.

   ```vba
   oFax.GenerateCompletionEvent = True
   ```

   `oFax` is the variable for the fax object you are creating.

4. Set the server to automatically generate a CompleteEvent for every completed fax by adding the following code to the Form_Load method of frmMain.

   ```vba
   EventsObj.WatchCompleteEvents = True
   ```
5. Define the action by specifying a function that will be called when the specified event occurs. The following example creates a message box containing the status of the fax and the name of the person to whom the fax was sent when the event handler finds a CompleteEvent.

```vbs
Private Sub MyFaxServerEventHandler_OnCompleteEvent(ByVal NewCompleteEvent As RFCOMAPILib.CompleteEvent)
    MsgBox "Found new complete event fax! Fax sent to " + NewCompleteEvent.Fax.ToName + ". Status = " + NewCompleteEvent.Fax.StatusDescription
End Sub
```

**ArchiveEvent**

The ArchiveEvent object is generated when a fax has completed sending. The IsArchiveEnable property in the User object must be set to True in order to archive a fax.

**Properties**

- **DeleteFax**
  This Boolean property specifies whether or not the fax should be deleted once it has been archived. True or 1 indicates it should be deleted; False or 0 indicates it should not be deleted.

- **Fax**
  Returns the Fax object associated with the Archive event.

- **FaxHandle**
  Returns the database handle of the Fax object associated with the ArchiveEvent.

- **User**
  Returns the User object associated with the ArchiveEvent.

- **UserID**
  Returns the ID of the User object associated with the ArchiveEvent.

**Methods**

None.
Event Handling and Event Objects

See Also

Archive, IsArchiveEnable, WatchArchiveEvents

CompleteEvent

The CompleteEvent object is generated when a fax has been sent if the GenerateCompletionEvent property in the Fax object is set to True.

Properties

DeleteFax
This Boolean property specifies whether or not the fax should be deleted once it has been completed. True or 1 indicates it should be deleted; False or 0 indicates it should not be deleted.

Fax
Returns the Fax object associated with the fired Complete event.

FaxHandle
Returns the database handle of the Fax object associated with the fired Complete event.

User
Returns the User object associated with the fired Complete event.

UserID
Returns the ID of the User object associated with the fired Complete event.

Methods

None.

See Also

GenerateCompletionEvent, OnCompleteEvent, WatchCompleteEvents
Events

The Events object controls which events get watched, dispatches found events, and retrieves events from the queue if available.

Properties

**Archive**
Retrieves an Archive event from the queue if available.

**Complete**
Retrieves a Complete event from the queue if available.

**Interval**
Specifies the time in seconds to wait before checking the event queue.

**Message**
Retrieves a Message event from the queue if available.

**Validate**
Retrieves a Validate event from the queue if available.

**WatchArchiveEvents**
This Boolean property specifies whether or not to check for Complete events. True or 1 checks for Complete events; False or 0 does not.

**WatchCompleteEvents**
This Boolean property specifies whether or not to check for Complete events. True or 1 checks for Complete events; False or 0 does not.

**WatchMessageEvents**
This Boolean property specifies whether or not to check for Message events. True or 1 checks for Message events; False or 0 does not.

**WatchNewFaxes**
This Boolean property specifies whether or not to watch for new faxes for a specific user. You must pass the User object as well as the Boolean value. True or 1 indicates the queue for the user will be checked; False or 0 indicates it will not.

**WatchValidateEvents**
This Boolean property specifies whether or not to check for Validate events. True or 1 checks for Validate events; False or 0 does not.
Event Handling and Event Objects

Methods

None.

See Also

ArchiveEvent, CompleteEvent, Events (property in the FaxServer object), MessageEvent, ValidateEvent

Example

The following code sets up the event handler to watch for archive events.

```vba
Private Sub Form_Load ()
    'Set Up Event Handler
    Set MyFaxServerEventHandler = MyFaxServer
    SetEventsObj=MyFaxServer.Events
    EventsObj.WatchArchiveEvents = True
EndSub
```

MessageEvent

A MessageEvent is generated when the user needs to be notified that something important happened to the fax. For example, if the user has the property SendNotifyOnSentSuccessfully set to True, then a message will be generated when a fax is sent successfully.

Properties

DeleteFax
This Boolean property specifies whether or not the fax should be deleted once the message has been sent. True or 1 indicates it should be deleted; False or 0 indicates it should not be deleted.

Fax
Returns the Fax object associated with the fired Message event.

FaxHandle
Returns the database handle of the Fax object associated with the fired Message event.

MessageEventType
Returns the type of message generated. This will be one of the valid MessageEventType enumerated values.
Table 10.1 MessageEventType Enumerated Values

<table>
<thead>
<tr>
<th>Name</th>
<th>Numeric equivalent</th>
</tr>
</thead>
<tbody>
<tr>
<td>meNetworkBroadcast</td>
<td>0</td>
</tr>
<tr>
<td>meCustom1</td>
<td>1</td>
</tr>
<tr>
<td>meCustom2</td>
<td>2</td>
</tr>
<tr>
<td>meCustom3</td>
<td>3</td>
</tr>
<tr>
<td>meCustom4</td>
<td>4</td>
</tr>
<tr>
<td>meCustom5</td>
<td>5</td>
</tr>
<tr>
<td>meCustom6</td>
<td>6</td>
</tr>
<tr>
<td>meCustom7</td>
<td>7</td>
</tr>
<tr>
<td>meCustom8</td>
<td>8</td>
</tr>
<tr>
<td>meCustom9</td>
<td>9</td>
</tr>
<tr>
<td>meCCMail</td>
<td>10</td>
</tr>
<tr>
<td>meMSMail</td>
<td>11</td>
</tr>
<tr>
<td>meGroupWise</td>
<td>12</td>
</tr>
<tr>
<td>meTRS</td>
<td>13</td>
</tr>
<tr>
<td>meCX3</td>
<td>14</td>
</tr>
<tr>
<td>mePager</td>
<td>15</td>
</tr>
<tr>
<td>meNotes</td>
<td>16</td>
</tr>
<tr>
<td>meExchange</td>
<td>17</td>
</tr>
<tr>
<td>meSMTP</td>
<td>18</td>
</tr>
</tbody>
</table>

**MessageText**
Returns the text of the notification.

**User**
Returns the User object associated with the fired MessageEvent.

**UserID**
Returns the ID of the User object associated with the fired MessageEvent.

**Methods**
None.
Event Handling and Event Objects

See Also

OnMessageEvent, SendNotifyOnIncompleteFirstTime, SendNotifyOnIncompletePeriodically, SendNotifyOnNoHoldForPreview, SendNotifyOnSendFailedWillRetry, SendNotifyOnSendFailure, SendNotifyOnSendingFirstTime, SendNotifyOnSendingPeriodically, SendNotifyOnSentSuccessfully, WatchMessageEvents

ValidateEvent

A ValidateEvent object is generated when NeedsFaxesApproved is True in the User object.

Properties

**BillingCode**
Returns the billing code associated with the fax, if any.

**Fax**
Returns the Fax object associated with the fired Validate event.

**FaxHandle**
Returns the database handle of the Fax object associated with the fired Validate event.

**ToFaxNumber**
Returns the destination fax number.

**User**
Returns the User object associated with the fired Validate event.

**UserID**
Returns the ID of the User object associated with the fired Validate event.

Methods

None.

See Also

OnValidateEvent, ValidateBillingCodes, WatchValidateEvents
Exercise 1

Add code to your project that does the following:

1. When a new fax is received the temporary TIF file associated with the fax is moved to the D:\faxarchive directory and named with a unique id.

2. Your code creates a comma delimited text file that is stored in the same directory and name with the same unique id but has a .txt extension. The fields in this comma delimited file are:
   - The owner id of the recipient
   - The remote id of the sender
# Appendix A

## RightFax Enumerations

### Table of RightFax Enumerations

#### AttachmentType

<table>
<thead>
<tr>
<th>Name</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>aNone</td>
<td>0</td>
</tr>
<tr>
<td>aFile</td>
<td>1</td>
</tr>
<tr>
<td>aLibraryDocument</td>
<td>2</td>
</tr>
<tr>
<td>aForm</td>
<td>3</td>
</tr>
<tr>
<td>aBillingCode</td>
<td>4</td>
</tr>
</tbody>
</table>

#### AutoForwardType

<table>
<thead>
<tr>
<th>Name</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>afFaxNumber</td>
<td>0</td>
</tr>
<tr>
<td>afUser</td>
<td>1</td>
</tr>
</tbody>
</table>

#### CommunicationProtocolType

<table>
<thead>
<tr>
<th>Name</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>cpNamedPipes</td>
<td>1</td>
</tr>
<tr>
<td>cplPXOS2</td>
<td>2</td>
</tr>
<tr>
<td>cpSPX</td>
<td>3</td>
</tr>
<tr>
<td>cpTCPPIP</td>
<td>4</td>
</tr>
<tr>
<td>cplPX</td>
<td>5</td>
</tr>
<tr>
<td>cpSecTCPPIP</td>
<td>6</td>
</tr>
<tr>
<td>cpSecSPX</td>
<td>7</td>
</tr>
</tbody>
</table>
RightFax Enumerations

**CreateObjectType**

<table>
<thead>
<tr>
<th>Name</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>coAttachment</td>
<td>0</td>
</tr>
<tr>
<td>coBillingCode</td>
<td>1</td>
</tr>
<tr>
<td>coCoverSheet</td>
<td>2</td>
</tr>
<tr>
<td>coDelegatee</td>
<td>3</td>
</tr>
<tr>
<td>coDelegator</td>
<td>4</td>
</tr>
<tr>
<td>coFax</td>
<td>5</td>
</tr>
<tr>
<td>coFolder</td>
<td>6</td>
</tr>
<tr>
<td>coForm</td>
<td>7</td>
</tr>
<tr>
<td>coGroup</td>
<td>8</td>
</tr>
<tr>
<td>coLibraryDocument</td>
<td>9</td>
</tr>
<tr>
<td>coPhoneItemGroup</td>
<td>10</td>
</tr>
<tr>
<td>coPhoneItemElement</td>
<td>11</td>
</tr>
<tr>
<td>coPrinter</td>
<td>12</td>
</tr>
<tr>
<td>coSignature</td>
<td>13</td>
</tr>
<tr>
<td>coUser</td>
<td>14</td>
</tr>
</tbody>
</table>

**EmailRouteFormType**

<table>
<thead>
<tr>
<th>Name</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>erfDefault</td>
<td>0</td>
</tr>
<tr>
<td>erfNone</td>
<td>1</td>
</tr>
<tr>
<td>erfStandard</td>
<td>2</td>
</tr>
<tr>
<td>erfAdvanced</td>
<td>3</td>
</tr>
</tbody>
</table>
## FaxErrorCodeType

<table>
<thead>
<tr>
<th>Name</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>fecNone</td>
<td>0</td>
</tr>
<tr>
<td>fecBusy</td>
<td>1</td>
</tr>
<tr>
<td>fecTransmissionError</td>
<td>2</td>
</tr>
<tr>
<td>fecPoorQuality</td>
<td>3</td>
</tr>
<tr>
<td>fecNoAnswer</td>
<td>4</td>
</tr>
<tr>
<td>fecBadFCS</td>
<td>5</td>
</tr>
<tr>
<td>fecBadConvert</td>
<td>6</td>
</tr>
<tr>
<td>fecMakeFCS</td>
<td>7</td>
</tr>
<tr>
<td>fecCantSchedule</td>
<td>8</td>
</tr>
<tr>
<td>fecUnknown</td>
<td>9</td>
</tr>
<tr>
<td>fecHuman</td>
<td>10</td>
</tr>
<tr>
<td>fecGroup2</td>
<td>11</td>
</tr>
<tr>
<td>fecLocalInUse</td>
<td>12</td>
</tr>
<tr>
<td>fecLineProblem</td>
<td>13</td>
</tr>
<tr>
<td>fecBadPaper</td>
<td>14</td>
</tr>
<tr>
<td>fecBadSignature</td>
<td>15</td>
</tr>
<tr>
<td>fecNoSignature Authorization</td>
<td>16</td>
</tr>
<tr>
<td>fecDiscarded</td>
<td>18</td>
</tr>
<tr>
<td>fecBadPhone</td>
<td>19</td>
</tr>
<tr>
<td>fecInvalidCode</td>
<td>21</td>
</tr>
<tr>
<td>fecBadCode</td>
<td>22</td>
</tr>
<tr>
<td>fecBadOCR</td>
<td>23</td>
</tr>
<tr>
<td>fecBadPrint</td>
<td>24</td>
</tr>
<tr>
<td>fecNoLibraryDocument Authorization</td>
<td>25</td>
</tr>
<tr>
<td>fecViewStar1</td>
<td>26</td>
</tr>
<tr>
<td>fecDisapproved</td>
<td>27</td>
</tr>
<tr>
<td>fecEmailDeliveryError</td>
<td>28</td>
</tr>
</tbody>
</table>
## RightFax Enumerations

### FaxHistoryType

<table>
<thead>
<tr>
<th>Name</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>FaxHistoryTransmission</td>
<td>0</td>
</tr>
<tr>
<td>FaxHistoryRoute</td>
<td>1</td>
</tr>
<tr>
<td>FaxHistoryPrint</td>
<td>2</td>
</tr>
<tr>
<td>FaxHistoryConversionError</td>
<td>3</td>
</tr>
<tr>
<td>FaxHistoryOCR</td>
<td>4</td>
</tr>
<tr>
<td>FaxHistoryNetForward</td>
<td>5</td>
</tr>
<tr>
<td>FaxHistoryApproval</td>
<td>6</td>
</tr>
<tr>
<td>FaxHistoryDisapproval</td>
<td>7</td>
</tr>
<tr>
<td>FaxHistoryFileRoute</td>
<td>8</td>
</tr>
<tr>
<td>FaxHistorySecureDoc</td>
<td>9</td>
</tr>
</tbody>
</table>

### FaxMarkType

<table>
<thead>
<tr>
<th>Name</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>fmViewed</td>
<td>0</td>
</tr>
<tr>
<td>fmUnViewed</td>
<td>1</td>
</tr>
<tr>
<td>fmPrinted</td>
<td>3</td>
</tr>
<tr>
<td>fmNotPrinted</td>
<td>4</td>
</tr>
<tr>
<td>fmReleased</td>
<td>6</td>
</tr>
<tr>
<td>fmHeld</td>
<td>6</td>
</tr>
<tr>
<td>fmBodyPrinted</td>
<td>7</td>
</tr>
<tr>
<td>fmApproved</td>
<td>8</td>
</tr>
<tr>
<td>fmDisapproved</td>
<td>9</td>
</tr>
<tr>
<td>fmGeneric1</td>
<td>10</td>
</tr>
<tr>
<td>fmNotGeneric1</td>
<td>11</td>
</tr>
<tr>
<td>fmGeneric2</td>
<td>12</td>
</tr>
<tr>
<td>fmNotGeneric2</td>
<td>13</td>
</tr>
</tbody>
</table>
### RightFax Enumerations

**FaxPriorityType**

<table>
<thead>
<tr>
<th>Name</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>fpNormal</td>
<td>0</td>
</tr>
<tr>
<td>fpLow</td>
<td>1</td>
</tr>
<tr>
<td>fpHigh</td>
<td>2</td>
</tr>
</tbody>
</table>

**FaxStatusType**

<table>
<thead>
<tr>
<th>Name</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>fsUnborn</td>
<td>0</td>
</tr>
<tr>
<td>fsNeedsFCS</td>
<td>1</td>
</tr>
<tr>
<td>fsNeedsConversion</td>
<td>2</td>
</tr>
<tr>
<td>fsNeedsToBeSent</td>
<td>3</td>
</tr>
<tr>
<td>fsInConversion</td>
<td>4</td>
</tr>
<tr>
<td>fsInSend</td>
<td>5</td>
</tr>
<tr>
<td>fsDoneOK</td>
<td>6</td>
</tr>
<tr>
<td>fsManualFCS</td>
<td>7</td>
</tr>
<tr>
<td>fsInSchedule</td>
<td>8</td>
</tr>
</tbody>
</table>

**GroupSFDType**

<table>
<thead>
<tr>
<th>Name</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>gsfdLinear</td>
<td>0</td>
</tr>
<tr>
<td>gsfdBalanced</td>
<td>1</td>
</tr>
</tbody>
</table>
**RightFax Enumerations**

### LibraryDocumentUsageType

<table>
<thead>
<tr>
<th>Name</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>lduNone</td>
<td>0</td>
</tr>
<tr>
<td>lduFOD</td>
<td>1</td>
</tr>
<tr>
<td>lduWeb</td>
<td>2</td>
</tr>
<tr>
<td>lduLAN</td>
<td>3</td>
</tr>
<tr>
<td>lduResetAll</td>
<td>0x0100</td>
</tr>
<tr>
<td>lduResetLastUsedDate</td>
<td>0x0200</td>
</tr>
</tbody>
</table>

### MessageEventType

<table>
<thead>
<tr>
<th>Name</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>meNetworkBroadcast</td>
<td>0</td>
</tr>
<tr>
<td>meCustom1</td>
<td>1</td>
</tr>
<tr>
<td>meCustom2</td>
<td>2</td>
</tr>
<tr>
<td>meCustom3</td>
<td>3</td>
</tr>
<tr>
<td>meCustom4</td>
<td>4</td>
</tr>
<tr>
<td>meCustom5</td>
<td>5</td>
</tr>
<tr>
<td>meCustom6</td>
<td>6</td>
</tr>
<tr>
<td>meCustom7</td>
<td>7</td>
</tr>
<tr>
<td>meCustom8</td>
<td>8</td>
</tr>
<tr>
<td>meCustom9</td>
<td>9</td>
</tr>
<tr>
<td>meCCMail</td>
<td>10</td>
</tr>
<tr>
<td>meMSMail</td>
<td>11</td>
</tr>
<tr>
<td>meGroupWise</td>
<td>12</td>
</tr>
<tr>
<td>meTRS</td>
<td>13</td>
</tr>
<tr>
<td>meCX3</td>
<td>14</td>
</tr>
<tr>
<td>mePager</td>
<td>15</td>
</tr>
<tr>
<td>meNotes</td>
<td>16</td>
</tr>
<tr>
<td>meExchange</td>
<td>17</td>
</tr>
<tr>
<td>meSMTP</td>
<td>18</td>
</tr>
</tbody>
</table>
### NotifyType

<table>
<thead>
<tr>
<th>Name</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>nNetworkBroadcast</td>
<td>0</td>
</tr>
<tr>
<td>nCustom1</td>
<td>1</td>
</tr>
<tr>
<td>nCustom2</td>
<td>2</td>
</tr>
<tr>
<td>nCustom3</td>
<td>3</td>
</tr>
<tr>
<td>nCustom4</td>
<td>4</td>
</tr>
<tr>
<td>nCustom5</td>
<td>5</td>
</tr>
<tr>
<td>nCustom6</td>
<td>6</td>
</tr>
<tr>
<td>nCustom7</td>
<td>7</td>
</tr>
<tr>
<td>nCustom8</td>
<td>8</td>
</tr>
<tr>
<td>nCustom9</td>
<td>9</td>
</tr>
<tr>
<td>nCCMail</td>
<td>10</td>
</tr>
<tr>
<td>nMSMail</td>
<td>11</td>
</tr>
<tr>
<td>nGroupWise</td>
<td>12</td>
</tr>
<tr>
<td>nTRS</td>
<td>13</td>
</tr>
<tr>
<td>nCX3</td>
<td>14</td>
</tr>
<tr>
<td>nPager</td>
<td>15</td>
</tr>
<tr>
<td>nNotes</td>
<td>16</td>
</tr>
<tr>
<td>nExchange</td>
<td>17</td>
</tr>
<tr>
<td>nSMTP</td>
<td>18</td>
</tr>
</tbody>
</table>

### OCRFormatType

<table>
<thead>
<tr>
<th>Name</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>ofSmartASCII</td>
<td>0</td>
</tr>
<tr>
<td>ofASCII</td>
<td>1</td>
</tr>
<tr>
<td>ofRTF</td>
<td>3</td>
</tr>
<tr>
<td>ofUseUserType</td>
<td>0xFFFF (See also AutoOCRFormat)</td>
</tr>
</tbody>
</table>
### OCRLayoutType

<table>
<thead>
<tr>
<th>Name</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>olWYSIWYG</td>
<td>0</td>
</tr>
<tr>
<td>olLeftJustified</td>
<td>1</td>
</tr>
<tr>
<td>olUseUserType</td>
<td>0xFFFF</td>
</tr>
</tbody>
</table>

### PhonelItemType

<table>
<thead>
<tr>
<th>Name</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>piElement</td>
<td>0</td>
</tr>
<tr>
<td>piGroup</td>
<td>1</td>
</tr>
</tbody>
</table>

### PrintDuplexType

<table>
<thead>
<tr>
<th>Name</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>pdDefault</td>
<td>0</td>
</tr>
<tr>
<td>pdSingle</td>
<td>1</td>
</tr>
<tr>
<td>pdLongEdge</td>
<td>2</td>
</tr>
<tr>
<td>pdShortEdge</td>
<td>3</td>
</tr>
</tbody>
</table>
### PrinterType

<table>
<thead>
<tr>
<th>Name</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>pNone</td>
<td>0</td>
</tr>
<tr>
<td>pLaserJet</td>
<td>1</td>
</tr>
<tr>
<td>pLaserJetII</td>
<td>2</td>
</tr>
<tr>
<td>pDeskJet</td>
<td>3</td>
</tr>
<tr>
<td>pLaserJetIII</td>
<td>4</td>
</tr>
<tr>
<td>pDeskJet500</td>
<td>5</td>
</tr>
<tr>
<td>pPostScriptLevel1</td>
<td>6</td>
</tr>
<tr>
<td>pLaserJet4</td>
<td>7</td>
</tr>
<tr>
<td>pPPDS</td>
<td>8</td>
</tr>
<tr>
<td>pXIP</td>
<td>9</td>
</tr>
<tr>
<td>pXeroxDCS20</td>
<td>10</td>
</tr>
<tr>
<td>pXeroxDCS35</td>
<td>11</td>
</tr>
<tr>
<td>pLaserJet3si</td>
<td>12</td>
</tr>
<tr>
<td>pLaserJet4si</td>
<td>13</td>
</tr>
<tr>
<td>pLaserJet5si</td>
<td>14</td>
</tr>
<tr>
<td>pATIImagePrinter</td>
<td>15</td>
</tr>
<tr>
<td>pLaserJet5siMopier</td>
<td>16</td>
</tr>
<tr>
<td>pXeroxN24</td>
<td>17</td>
</tr>
<tr>
<td>pXeroxDCS220</td>
<td>18</td>
</tr>
<tr>
<td>pXeroxDCS230</td>
<td>19</td>
</tr>
<tr>
<td>pXeroxDCS240</td>
<td>20</td>
</tr>
<tr>
<td>pXeroxDCS255</td>
<td>21</td>
</tr>
<tr>
<td>pXeroxDCS265</td>
<td>22</td>
</tr>
<tr>
<td>pXeroxRicoh250</td>
<td>23</td>
</tr>
<tr>
<td>pXeroxLaserJet8000</td>
<td>24</td>
</tr>
<tr>
<td>pXeroxGDI</td>
<td>25</td>
</tr>
<tr>
<td>pXeroxDCS460</td>
<td>26</td>
</tr>
<tr>
<td>pXeroxDCS470</td>
<td>27</td>
</tr>
<tr>
<td>pXeroxDCS480</td>
<td>28</td>
</tr>
</tbody>
</table>
## RightFax Enumerations

### PrintJobType

<table>
<thead>
<tr>
<th>Name</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>pjPCL</td>
<td>1</td>
</tr>
<tr>
<td>pjPostScript</td>
<td>2</td>
</tr>
<tr>
<td>pjPCL2</td>
<td>3</td>
</tr>
<tr>
<td>pjPostScript2</td>
<td>4</td>
</tr>
<tr>
<td>pjCVL</td>
<td>5</td>
</tr>
</tbody>
</table>

### PrintOutputBinType

<table>
<thead>
<tr>
<th>Name</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>poBinDefault</td>
<td>0</td>
</tr>
<tr>
<td>poBinUpper</td>
<td>1</td>
</tr>
<tr>
<td>poBinLower</td>
<td>2</td>
</tr>
<tr>
<td>poBinFinisher</td>
<td>3</td>
</tr>
</tbody>
</table>

### PrintPageSourceType

<table>
<thead>
<tr>
<th>Name</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>ppsDefault</td>
<td>0</td>
</tr>
<tr>
<td>ppsUpperOrTray1</td>
<td>1</td>
</tr>
<tr>
<td>ppsLowerOrTray2</td>
<td>2</td>
</tr>
<tr>
<td>ppsManual</td>
<td>3</td>
</tr>
<tr>
<td>ppsTray3</td>
<td>4</td>
</tr>
<tr>
<td>ppsMPTrayOrTray4</td>
<td>5</td>
</tr>
<tr>
<td>ppsHighCapacityOrTray5</td>
<td>6</td>
</tr>
</tbody>
</table>

### PrintResolutionType

<table>
<thead>
<tr>
<th>Name</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>prHigh</td>
<td>0</td>
</tr>
<tr>
<td>prMedium</td>
<td>1</td>
</tr>
<tr>
<td>prLow</td>
<td>2</td>
</tr>
</tbody>
</table>
### PrintSizeType

<table>
<thead>
<tr>
<th>Name</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>psDefault</td>
<td>0</td>
</tr>
<tr>
<td>psLetter</td>
<td>1</td>
</tr>
<tr>
<td>psLegal</td>
<td>2</td>
</tr>
<tr>
<td>psA4</td>
<td>3</td>
</tr>
<tr>
<td>psLetterLegal</td>
<td>4</td>
</tr>
<tr>
<td>psA4Legal</td>
<td>5</td>
</tr>
</tbody>
</table>

### ReceiveNotifyType

<table>
<thead>
<tr>
<th>Name</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>rnNone</td>
<td>0</td>
</tr>
<tr>
<td>rnOnce</td>
<td>1</td>
</tr>
<tr>
<td>rnPeriodically</td>
<td>2</td>
</tr>
</tbody>
</table>

### RequiredFieldIndexType

<table>
<thead>
<tr>
<th>Name</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>rfiToName</td>
<td>0</td>
</tr>
<tr>
<td>rfiToFaxNumber</td>
<td>1</td>
</tr>
<tr>
<td>rfiToVoiceNumber</td>
<td>2</td>
</tr>
<tr>
<td>rfiToCompany</td>
<td>3</td>
</tr>
<tr>
<td>rfiToCityState</td>
<td>4</td>
</tr>
<tr>
<td>rfiFromName</td>
<td>5</td>
</tr>
<tr>
<td>rfiFromPersonalVoiceNumber</td>
<td>6</td>
</tr>
<tr>
<td>rfiFromPersonalFaxNumber</td>
<td>7</td>
</tr>
<tr>
<td>rfiFromGeneralVoiceNumber</td>
<td>8</td>
</tr>
<tr>
<td>rfiFromGeneralFaxNumber</td>
<td>9</td>
</tr>
<tr>
<td>rfiBillCode1</td>
<td>10</td>
</tr>
<tr>
<td>rfiBillCode2</td>
<td>11</td>
</tr>
</tbody>
</table>
RightFax Enumerations

### RequiredFieldType

<table>
<thead>
<tr>
<th>Name</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>rfNotRequired</td>
<td>0</td>
</tr>
<tr>
<td>rfSend</td>
<td>1</td>
</tr>
<tr>
<td>rfReceive</td>
<td>2</td>
</tr>
<tr>
<td>rfAlways</td>
<td>3</td>
</tr>
</tbody>
</table>

### RouteType

<table>
<thead>
<tr>
<th>Name</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>rRightFax</td>
<td>0</td>
</tr>
<tr>
<td>rCCMail</td>
<td>1</td>
</tr>
<tr>
<td>rMSMail</td>
<td>2</td>
</tr>
<tr>
<td>rFile</td>
<td>4</td>
</tr>
<tr>
<td>rOCR</td>
<td>5</td>
</tr>
<tr>
<td>rGroupWiseMail</td>
<td>6</td>
</tr>
<tr>
<td>rNotes</td>
<td>7</td>
</tr>
<tr>
<td>rXRoute</td>
<td>8</td>
</tr>
<tr>
<td>rTRS</td>
<td>9</td>
</tr>
<tr>
<td>rCallXPress</td>
<td>10</td>
</tr>
<tr>
<td>rExchange</td>
<td>11</td>
</tr>
<tr>
<td>rSMTP</td>
<td>12</td>
</tr>
<tr>
<td>rANI</td>
<td>13</td>
</tr>
<tr>
<td>rSAP1</td>
<td>14</td>
</tr>
<tr>
<td>rSAP2</td>
<td>15</td>
</tr>
<tr>
<td>rSAP3</td>
<td>16</td>
</tr>
<tr>
<td>rSAP4</td>
<td>17</td>
</tr>
<tr>
<td>rSAP5</td>
<td>18</td>
</tr>
<tr>
<td>rSAP6</td>
<td>19</td>
</tr>
<tr>
<td>rSAP7</td>
<td>20</td>
</tr>
<tr>
<td>rSAP8</td>
<td>21</td>
</tr>
</tbody>
</table>
# RightFax Enumerations

## RouteType (continued)

<table>
<thead>
<tr>
<th>Name</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>rSAP9</td>
<td>22</td>
</tr>
<tr>
<td>rSAP10</td>
<td>23</td>
</tr>
<tr>
<td>rSAP11</td>
<td>24</td>
</tr>
<tr>
<td>rSAP12</td>
<td>25</td>
</tr>
<tr>
<td>rSAP13</td>
<td>26</td>
</tr>
<tr>
<td>rSAP14</td>
<td>27</td>
</tr>
<tr>
<td>rSAP15</td>
<td>28</td>
</tr>
<tr>
<td>rSAP16</td>
<td>29</td>
</tr>
</tbody>
</table>

## ServerSpecialType

<table>
<thead>
<tr>
<th>Name</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>ssDelear</td>
<td>9999</td>
</tr>
<tr>
<td>ssTampered</td>
<td>9998</td>
</tr>
</tbody>
</table>

## ServerType

<table>
<thead>
<tr>
<th>Name</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>sOS2</td>
<td>3313</td>
</tr>
<tr>
<td>sNT</td>
<td>3314</td>
</tr>
</tbody>
</table>

## TimeZoneInfoType

<table>
<thead>
<tr>
<th>Name</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>tzInvalid</td>
<td>0</td>
</tr>
<tr>
<td>tzValid</td>
<td>1</td>
</tr>
<tr>
<td>tzStandardTime</td>
<td>2</td>
</tr>
<tr>
<td>tzDaylightSavingsTime</td>
<td>3</td>
</tr>
</tbody>
</table>
**RightFax Enumerations**

### TransmissionFaxHistoryBoardType

<table>
<thead>
<tr>
<th>Name</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>tfhbtBrooktrout</td>
<td>1</td>
</tr>
<tr>
<td>tfhbtGammalink</td>
<td>2</td>
</tr>
<tr>
<td>tfhbtBrooktroutNew</td>
<td>3</td>
</tr>
</tbody>
</table>

### TransmissionFaxHistoryTerminationStatusType

<table>
<thead>
<tr>
<th>Name</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>tfhtstSuccessReceive</td>
<td>0x0</td>
</tr>
<tr>
<td>tfhtstBusy1</td>
<td>0x1</td>
</tr>
<tr>
<td>tfhtstBusy2</td>
<td>0x2</td>
</tr>
<tr>
<td>tfhtstReorderBusy</td>
<td>0x3</td>
</tr>
<tr>
<td>tfhtstRecall</td>
<td>0x4</td>
</tr>
<tr>
<td>tfhtstConfirm</td>
<td>0x5</td>
</tr>
<tr>
<td>tfhtstBlocked</td>
<td>0x6</td>
</tr>
<tr>
<td>tfhtstRing1</td>
<td>0x8</td>
</tr>
<tr>
<td>tfhtstRing2</td>
<td>0x9</td>
</tr>
<tr>
<td>tfhtstHuman</td>
<td>0x10</td>
</tr>
<tr>
<td>tfhtstAnswer</td>
<td>0x11</td>
</tr>
<tr>
<td>tfhtstDialtone</td>
<td>0x12</td>
</tr>
<tr>
<td>tfhtstSITDetect</td>
<td>0x13</td>
</tr>
<tr>
<td>tfhtstSITNOCIR</td>
<td>0x14</td>
</tr>
<tr>
<td>tfhtstTransmissionError</td>
<td>0x15</td>
</tr>
<tr>
<td>tfhtstHangNoLoopCurrent</td>
<td>0x16</td>
</tr>
<tr>
<td>tfhtstCallColide</td>
<td>0x17</td>
</tr>
<tr>
<td>tfhtstDialNoLoopCurrent</td>
<td>0x18</td>
</tr>
<tr>
<td>tfhtstRingNoAnswer</td>
<td>0x19</td>
</tr>
<tr>
<td>tfhtstGroup2Detected</td>
<td>0x1a</td>
</tr>
<tr>
<td>tfhtstQuiet</td>
<td>0x1c</td>
</tr>
<tr>
<td>tfhtstFaxMachine</td>
<td>0x20</td>
</tr>
<tr>
<td>tfhtstLocalInUse</td>
<td>0x21</td>
</tr>
<tr>
<td>tfhtstSilence</td>
<td>0x22</td>
</tr>
<tr>
<td>tfhtstUnknown</td>
<td>0x23</td>
</tr>
</tbody>
</table>
### UserRouteFormatType

<table>
<thead>
<tr>
<th>Name</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>urfDCX</td>
<td>0</td>
</tr>
<tr>
<td>urfPCX</td>
<td>1</td>
</tr>
<tr>
<td>urfTiffGroup3</td>
<td>2</td>
</tr>
<tr>
<td>urfTiffGroup4</td>
<td>3</td>
</tr>
<tr>
<td>urfGIF</td>
<td>4</td>
</tr>
<tr>
<td>urfPDF</td>
<td>5</td>
</tr>
<tr>
<td>urfPDFWithThumbnails</td>
<td>6</td>
</tr>
<tr>
<td>urfCPC</td>
<td>7</td>
</tr>
<tr>
<td>urfRFX</td>
<td>8</td>
</tr>
<tr>
<td>urfPNG</td>
<td>9</td>
</tr>
</tbody>
</table>