

# FaxPack™ 7.0 Plug-in for REALbasic & 4-Sight FAX



© 2007 Soft Solutions, Inc. All Rights Reserved.

Published World-Wide by Soft Solutions, Inc.

Last updated on August 7th, 2007.

# Disclosure

By installing this package, you are agreeing to be bound to the terms of this agreement.

## Soft Solutions, Inc. Software License

1. GRANT OF LICENSE. Soft Solutions grants to you, the Developer, the right to use the Soft Solutions software program FaxPack (the "SOFTWARE") on a single computer (i.e., with a single CPU), or on a LICENSED COMPUTER NETWORK. Each concurrent user on a network will be governed under the Deployment license purchased. Soft Solutions, Inc. as Licensor, grants to you, the LICENSEE, a non-exclusive, non-transferable right to use this Software subject to the terms of the license as described in the following sections:
  - A. You may make backup copies of the Software for your use provided they bear the Soft Solutions, Inc copyright notice.
  - B. You may use this software in a manner consistent with the license purchased. Please refer to the terms of your license at the 4-Sight FAX web site, located at <http://www.4sightfax.com>.
2. COPYRIGHT. The SOFTWARE is owned by Soft Solutions and is protected by United States copyright laws and international treaty provisions. Therefore, you must treat the SOFTWARE like any other copyrighted material (e.g., a book or musical recording) except that you may either (a) make one copy of the SOFTWARE solely for backup or archival purposes, or (b) transfer the SOFTWARE to a single hard disk provided you keep the original solely for backup or archival purposes. You may not copy the written materials accompanying the SOFTWARE.
3. OTHER RESTRICTIONS. You may not rent, lease or transfer the SOFTWARE. You may not reverse engineer, decimalize, or decompile the SOFTWARE.

LIMITED WARRANTY. Soft Solutions, Inc. warrants that (a) the SOFTWARE will perform substantially in accordance with the accompanying written materials for a period of ninety (90) days from the date of receipt. Any implied warranties on the SOFTWARE are limited to ninety (90) days and one (1) year, respectively. Some states do not allow limitations on duration of an implied warranty, so the above limitation may not apply to you.

CUSTOMER REMEDIES. Soft Solutions, Inc. entire liability and your exclusive remedy shall be, at Soft Solutions option, either: (a) return of the price paid or (b) repair or replacement of the SOFTWARE that does not meet Soft Solutions Limited Warranty. This Limited Warranty is void if failure of the SOFTWARE has resulted from accident, abuse, or misapplication. Any replacement SOFTWARE will be warranted for the remainder of the original warranty period of thirty (30) days, whichever is longer.

NO OTHER WARRANTIES. Soft Solutions disclaims all other warranties, either express or implied, including but not limited to implied warranties of merchantability and fitness for a particular purpose, with respect to the SOFTWARE, the accompanying written materials, and any accompanying hardware. This limited warranty gives you specific legal rights. You may have other rights, which vary by locale.

NO LIABILITY FOR CONSEQUENTIAL DAMAGES. In no event shall Soft Solutions or its suppliers be liable for any damages whatsoever (including, without limitation, damages for loss of business profits, business interruption, loss of business information, or other pecuniary loss) arising out of the use of or inability to use this Soft Solutions product, even if Soft Solutions has been advised of the possibility of such damages. Because some states do not allow the exclusion or limitation of liability for consequential or incidental damages, the above limitation may not apply to you.

Governing Law. This entire agreement shall be governed by the laws of the State of Georgia within the USA.

FaxPack is a trademark of Soft Solutions, Inc. Apple is a registered trademark of Apple Computer, Inc.

REALbasic and Filemaker Server are registered trademarks of REALbasic, Inc.

Soft Solutions, Inc.

2900 Chamblee Tucker Road

Building 12, Suite 200

Atlanta, GA 30341 USA

# TABLE OF CONTENTS

Introduction.....	1
FaxPack Version 7.0 .....	2
Command Overview .....	3
Using the FaxPack plug-in.....	3
Plug-in Classes .....	3
Server Selection and Login .....	3
Switching to the 4-Sight Fax Printer .....	3
Creating the Fax / Email message .....	4
Adding Message Attachments.....	4
Completing the Message.....	4
Message Details and Retrieval.....	5
Synchronizing REALbasic and the 4-Sight FAX Server Address Book .....	5
Managing Junk Faxes .....	5
Version Command.....	5
Installation .....	6
What to Download .....	6
What to Install.....	6
Contents of the Installation .....	6
4-Sight FAX Server and Configuration .....	7
Methods and Functions (Command Descriptions).....	8
Message Control .....	8
FaxPack.Version.....	8
FaxPack.StartMessage.....	8
FaxPack.AddRecipient.....	11
FaxPack.AddEnclosure.....	14
FaxPack.AddComment .....	14
FaxPack.SendMessage .....	15
FaxPack.AbandonMessage .....	16
FaxPack.GetItemStatus .....	16
Printer Control .....	18
FaxPack.SwitchPrinter.....	18
FaxPack.PreparePrint.....	18
Server Control .....	20
FaxPack.ServerLookup.....	20
FaxPack.ServerLogin.....	21
FaxPack.GetCoverPages.....	22
FaxPack.GetQueueListing .....	22
FaxPack.GetFaxTiff .....	25
Image Control.....	26
FaxPack.GetTiffInfo .....	26
FaxPack.ConvertTiff .....	26

Address Book .....	28
FaxPack.GetAddressBook .....	28
FaxPack.GetContact .....	29
FaxPack.ModifyContact .....	29
FaxPack.AddContact .....	30
FaxPack.DeleteContact .....	31
Block List .....	32
FaxPack.AddToBlockList .....	32
FaxPack.DeleteFromBlockList .....	33
Command Summary (Methods and Functions) .....	34
Properties .....	35
FaxPack Properties .....	35
Constants .....	36
Priority .....	36
DestinationKind .....	36
otherFields.....	36
Resolution .....	37
convertType.....	37
Classes.....	38
CoverPageList.....	38
ServerItem.....	38
ServerList .....	39
ItemStatus .....	40
ContactItem .....	41
Error Codes .....	44
Error Handling .....	46
Frequently Asked Questions (FAQ) .....	47
4-Sight FAX Sample Configuration.....	48
Contact Information .....	48
Index.....	49

## TABLE OF FIGURES

<b>Figure 1:</b> New User Setup .....	7
<b>Figure 2:</b> Cover Page Setup .....	10
<b>Figure 3:</b> Smart Dialing Setup on 4-Sight FAX Server.....	13
<b>Figure 4:</b> Area Code Rules configuration.....	13
<b>Figure 5:</b> Address Book Privileges.....	28
<b>Figure 6:</b> Login and Access .....	32
<b>Figure 7:</b> 4-Sight FAX Sample Configuration.....	48

# Introduction

Congratulations on your choice of FaxPack! An industrial strength fax solution, FaxPack is used in applications worldwide demanding direct and seamless database faxing and email capabilities. With FaxPack, REALbasic Developers can automate direct messaging capabilities for their users in their REALbasic application without the need for the 4-Sight FAX client.

Below is a short list of how FaxPack is being used in database applications (for other development environments) to simplify and improve the lives of its users:

- Emergency weather fax broadcasting
- Stock alert notification system
- Physician's stress and EKG fax notification
- Hospital information system w/ fax integration
- Order entry / pick list fax notification
- Hotel reservations fax confirmation
- Purchase order request system
- Patient medical record requests w/ automated fax-back and email delivery
- Voice-activated auto repair schematics system providing fax-back to auto service and repair centers
- Physician's prescription (Rx) fax-in system
- Software company invoice fax system
- Restaurant weekly lunch menu specials fax blast software
- Newspaper classified advertisements with fax / email review & confirmation

# FaxPack Version 7.0

FaxPack Versions 7 provides the following commands and features enabling database developers to:

- Retrieve details for entire fax / email queue listings.
- Retrieve individual faxes as TIFF files.
- Convert fax TIFF files into any of 5 alternative file formats.
- Have direct access to 4-Sight FAX Servers Smart dialing capabilities.
- Utilize permanent unique numbers for each fax (incoming and/or outgoing).
- Extend fax merge capabilities with access to 17 fields from the server address book.
- Add, update or delete FAX server-based address books directly from REALbasic.
- Download the entire 4-Sight FAX Server Address book to a local text file.
- Add or remove company fax numbers from the Fax Server Block list (feature).
- Send either fax or email messages (or both concurrently).
- Have access to any fax or email message status.

Under version 7, the 4-Sight FAX Server software uses a permanent unique numbering system. This permits database applications to create, retrieve and delete faxes using permanent, 100% accurate queue information.

# Command Overview

## *Using the FaxPack plug-in*

When installed into the "plugins" folder next to the REALbasic IDE, the FaxPack plug-in makes a new module called "FaxPack" available. This contains most of the methods of the plug-in, in protected scope, so they can be accessed by prefixing them with "FaxPack." For example, to get the plug-in version, you would call "FaxPack.Version".

## *Plug-in Classes*

The plug-in also adds several classes, as follows: **CoverPageList**, which Holds a list of cover page names, **ServerItem**, which holds the name and TCP/IP information of a 4-Sight FAX Server, **ServerList**, which holds a list of ServerItem instances, **ItemStatus**, which holds the status, state and other related information for a particular fax or email message, and **ContactItem**, which holds the information for a contact from the server address book.

## *Server Selection and Login*

The FAX server can be identified explicitly using the **FaxPack.ServerLogin** command along with a User name and password. The **FaxPack.ServerLookup** command can be used to search for a FAX server on the local network.

The **FaxPack.ServerLogin** command can optionally validate the information. The **FaxPack.ServerLogin** parameter values are made available to over a half dozen FaxPack commands that require login access dynamically. Each of these commands logs in and logs out of the server using the **FaxPack.ServerLogin** parameter values. A valid user name and password are checked against the 4-Sight FAX Server before any faxes are created and are a login requirement with version 7.



**IMPORTANT:** The 4-Sight FAX server permits a REALbasic application using FaxPack Version 7 and a 4-Sight FAX v7 client to run concurrently on the same machine with the same user[name] using the same TCP/IP address. With FaxPack for REALbasic, users are logged in as certain FaxPack commands are executed and logged out as FaxPack commands are completed. With the 4-Sight FAX Client software, however, users login and remain logged in until they log out.

## *Switching to the 4-Sight Fax Printer*

The **FaxPack.SwitchPrinter** command allows you to procedurally alternate between the 4-Sight FAX printer driver and the user's selected printer, without opening the Macintosh OS X "Print Center" or the Windows printer Control Panel. When running under Mac OS X, version 10.3 or later must be installed in order to use the "**FaxPack.SwitchPrinter**" command.

## ***Creating the Fax / Email message***

A fax / email message (message) is built with several FaxPack calls. To begin, the REALbasic developer should call the **FaxPack.StartMessage** command to provide basic sender and sending information. A “message” in this context refers to either a fax or email, generated through a REALbasic print job for transmission to one or more recipients. This command allocates the memory to store information about the message and its contents.

One or more destination addresses are then specified for the current message by calling the **FaxPack.AddRecipient** command. You must issue this call at least once, to specify at least one destination address, but there is no limit to the number of additional calls/destinations that may be specified. This command allocates the memory to store information about the message recipients, their fax numbers and associated information.

The **FaxPack.GetCoverPages** command provides the REALbasic Developer with the names of all the Fax Server cover pages stored on the 4-Sight FAX Server. Use this command to present the end-user with a list of cover pages they can choose from for sending with their fax. The **FaxPack.AddComment** command can optionally be used to specify a text comment for the fax cover page of the current fax message.

The **FaxPack.PreparePrint** command communicates directly with the 4-Sight FAX v7 print driver. This command tells FaxPack to capture the subsequent REALbasic print job into a temporary Tiff file for transmission to the 4-Sight FAX server. **FaxPack.PreparePrint** should be executed after the **FaxPack.StartMessage** command.

## ***Adding Message Attachments***

A REALbasic print job can be added as an attachment to the current fax message that began with **FaxPack.StartMessage**. The additional command order to do so is as follows: 1) Switch the default printer to 4-Sight Fax using **FaxPack.SwitchPrinter**. 2) Call the **FaxPack.PreparePrint** command. 3) Print from REALbasic, 4) Repeat steps 2-4 for more attachments. 5) Switch the default printer back using **FaxPack.SwitchPrinter**. Note that this technique assumes REALbasic will print to the default printer.

## ***Completing the Message***

Finally, the **FaxPack.SendMessage** command is called to build the message and send it to the 4-Sight FAX server for final processing and transmission. Alternatively, the **FaxPack.AbandonMessage** command can be used to dispose of the current message without transmitting it. The **FaxPack.SendMessage** command returns a unique “message ID” from the 4-Sight FAX v7 fax server. The same “message ID” returned within the **FaxPack.SendMessage** command can later be used for retrieving the fax status via the **FaxPack.GetItemStatus** command.



**IMPORTANT:** Every call to **FaxPack.StartMessage** must be balanced with a final call to either **FaxPack.SendMessage** or **FaxPack.AbandonMessage**.

## ***Message Details and Retrieval***

The **FaxPack.GetItemStatus** command allows the user to retrieve the fax status, date and time sent, etc. for any fax. The **FaxPack.GetFaxTiff** allows the REALbasic application to retrieve any fax file and is most useful for databases with viewing capabilities. The **FaxPack.GetTiffInfo** command provides the developer with the number of pages for the fax just retrieved. The **FaxPack.ConvertTiff** command converts one or more pages from a Fax Tiff file into any of 5 different file formats, including PDF, Pict, PNG, BMP and Tiff. When using the **FaxPack.ConvertTiff** command, files are converted and saved to Tiff file.

## ***Synchronizing REALbasic and the 4-Sight FAX Server Address Book***

The **FaxPack.GetContact** command provides the REALbasic developer a tool for retrieving 22 fields of information for any Contact within the FAX Server address. The REALbasic Developer accomplishes this by providing the unique ID of the Contract address book entry. This unique Contact ID is first provided by FaxPack after the Developer issues a **FaxPack.AddContact** command which adds the Contact to the 4-Sight FAX Server address book with up to 22 fields of data. Developers can modify Server address book capabilities with the **FaxPack.ModifyContact** command. Lastly, the **FaxPack.GetAddressBook** will retrieve all contact records from the Server Address book and return them to a text file, as specified by the Developer. This is useful for companies that want their 4-Sight FAX Server Address books synchronized with their REALbasic database.



**IMPORTANT:** When synchronizing your REALbasic database with the 4-Sight FAX Server address book, it will be essential to store the ContactID of the Address Book Contact within the REALbasic database.

## ***Managing Junk Faxes***

The 4-Sight FAX Server maintains a list of incoming fax numbers to block. This feature is available to prevent the receiving of unwanted faxes from specific fax numbers or companies.

The FaxPack commands for adding and removing fax numbers and companies from the “Block” Fax List are **FaxPack.AddToBlockList** and **FaxPack.DeleteFromBlockList**.

## ***Version Command***

The FaxPack.Version command returns the string of the current FaxPack version number.

# Installation

## *What to Download*

Download either:

1. FaxPack for REALbasic (Macintosh).
  2. FaxPack for REALbasic (Windows).
- Based on the version of REALbasic you are operating, please pick the appropriate download file based on the target platform. Please see <http://www.4sightfax.com/faxpackrb.html>. A Demonstration application is available for the REALbasic developer. This sample program outlines the syntax, while providing a rich sample of command examples along with a list of error codes and their use.

## *What to Install*

FaxPack requires the installation of the 4-Sight FAX Server and, at minimum, the print driver portion of the 4-Sight FAX version 7 client. An installer is provided for both the 4-Sight FAX v7 print driver for Macintosh OS X and the Windows print driver.

Installation of the FaxPack plug-in is achieved by copying the plug-in file to the REALbasic plugins folder.

Under Mac OS X, copy the "FaxPack.rbx" file into the REALbasic plugins folder.

Under Windows, copy the "FaxPack.rbx" file into the REALbasic plugins folder.

## *Contents of the Installation*

The FaxPack folder and its contents are contained within 2 separate compacted files and are available for download from the 4-Sight FAX web site. There is one download for Macintosh and one download for Windows.

The following items are contained in the FaxPack 7.0 Folder:

The FaxPack Demo Version 7 Folder contains:

- FaxPack demo application files
- FaxPack Order form
- FaxPack plugin files for the appropriate REALbasic plugins folder
- FaxPack Manual
- Installer for the 4-Sight FAX print driver (for installation on client machines and machines where FaxPack applications operate). Must install locally!
- The FaxPack Demo application contains freely distributable source code.

FaxPack developers should ensure the most current 4-Sight FAX print driver is installed first.

## 4-Sight FAX Server and Configuration

### New User Setup:



The 4-Sight FAX Server New User / Edit User dialog must be configured for each user who intends to fax and email using FaxPack (as well as from the 4-Sight FAX Client). First, notice the Username and Password fields. These are used by the FaxPack.ServerLogin command. Note the “Full Queue Access” checkbox, which enables users to download their faxes or all faxes from the queue within the FaxPack.GetQueueListing command. Next, note the Block List options, which are utilized with the FaxPack.AddToBlockList and FaxPack.DeleteFromBlockList commands. The broadcast fax checkbox allows Users to fax or email to more than 25 recipients per transmission. Failure to check this would prevent a User from faxing to more than 25 recipients at a time. The Server Address Book Privileges provide access and use of 5 Address Book commands available in this area. For Users who intend to send emails as well as faxes, the Email suite options must also be established. Normally, this dialog is accessed and used by the 4-Sight FAX Administrator, but it also helps when the REALbasic Developer can specifically ask for what their Users need.

**New User**

Name:

Password:

Group Membership:

**Login and Access:**

- Administrator access to server
- Full queue access
- Allow automatic login
- Allow username alias
- Enforce group name as public
- Allow add to Block List
- Allow delete from Block List

**Email Suite:**

- Enable email suite
- Enable email forwarding

Incoming email address:

Outgoing email address:

**Dialing and Routing:**

- Enable In-Bound Routing

User fax number:

Specify Outgoing Fax Line:

- Allow Client to specify

**Faxing Privileges:**

- Priority fax
- Broadcast fax
- Private fax

**Server Address Book Privileges:**

- Add Address Book Contact
- Modify Address Book Contact
- Delete Address Book Contact

**Printing:**

- Use General Settings
- No Printing
- Direct Printing

**Phone Card:**

- Before fax number
- After fax number
- Allow Client to specify

Figure 1: New User Setup

# Methods and Functions (Command Descriptions)

## *Message Control*

---

### **FaxPack.Version**

**Summary:**

Returns FaxPack plug-in version.

**Purpose:**

The FaxPack.Version command returns the version number of the FaxPack plug-in.

**Syntax:**

result=**FaxPack.Version**

Part	Type	Description
Result	String	Version of the FaxPack plug-in

**Example:**

```
Dim version as String  
Version=FaxPack.Version  
MsgBox("FaxPack version is "+version)
```

---

### **FaxPack.StartMessage**

**Summary:**

First method to call to begin building a new message.

**Purpose:**

The FaxPack.StartMessage method begins the process of building a new message. It allocates memory for storing the information to be sent to the Fax Server for the message, and allows you to define the information for the fax cover page.

Always begin the process of creating a message with a call to FaxPack.StartMessage, and then use other FaxPack methods to define the content of the message. The message is then either sent to the 4-Sight FAX Server with a call to FaxPack.SendMessage or cancelled with a call to FaxPack.AbandonMessage. Please note; the 4-Sight FAX Server may require the "from" parameter to be a name from amongst its list of Users, otherwise the Fax message will be rejected.



After you call FaxPack.StartMessage, a corresponding FaxPack.SendMessage or FaxPack.AbandonMessage call must be issued prior to any additional FaxPack.Start-

Message calls being issued. Either of these commands will discard any outstanding message information. REALbasic developers should call the FaxPack.AbandonMessage whenever an uncorrectable error occurs.

After calling FaxPack.StartMessage, test FaxPack.ErrorCode against zero (where zero is successful). Or, if the FaxPack.RaiseOption is ON your Catch statement will be executed if FaxPack.StartMessage failed.

**Syntax:**

**FaxPack.StartMessage**(From, Subject, CoverPage, Priority, SendDateTime)

Part	Type	Description
From	String	Whom the message is from. This string will appear in the "From" field on the fax header page.  <b>IMPORTANT:</b> This parameter requires a valid User name as defined in the 4-Sight FAX Server list of User names.
Subject	String	Message subject. A string expression containing a brief description of the fax content that will appear in the "Subject" field on the fax header page
CoverPage	UInt16	The cover page number. The 4-Sight FAX v7 server provides up to 14 unique cover pages. A value of 1 will select the first header page, 2 will select the second, and so on. A value of zero indicates no header page is required.  <b>IMPORTANT:</b> If the FaxPack.AddComment method is used, yet no cover page is specified here (a 0), the default cover page will be used when this option is established as the default within the 4-Sight FAX Server preferences. Please refer to the 4-Sight FAX Server Administrator manual for more on default cover pages.
Priority	UInt16	Message priority: 1 = Normal, 2 = Bulk, 3 = ASAP, 4 = Important, 5 = Urgent.
SendDateTime	Date	Local date & time of when you are requesting the 4-Sight FAX Server to fax or email the message. Passing Nil will indicate the current local date & time.

**Example:**

```

FaxPack.StartMessage("Wally", "Fax Test", 1, 5, Nil)
if FaxPack.ErrorCode=0 then
    MsgBox("FaxPack.StartMessage was successful")
else
    MsgBox("FaxPack.StartMessage failed, error#" + Str(FaxPack.ErrorCode))
end If

```

## Cover Page Setup:



The 4-Sight FAX option checked below will force a cover page to print when none is specified. If the user does not choose a cover page within FaxPack.StartMessage, yet issues the FaxPack.AddComment with comments, a cover page will be used. The cover page chosen is determined by the settings on the dialog below. If the checkbox in the Cover Page Configuration is checked, then the Cover Page specified in Profile1 will be used. If the checkbox within the Cover Page Configuration is NOT checked, then the Cover Page built into 4-Sight FAX Server's internal resources will be used.

**Cover Page Configuration**

Use Profile1 when 'None' is specified

Resolution:  Standard  Fine

Profile1: Standard Full Page

Profile2: Standard Full Page

Profile3: Standard Full Page

Profile4: Standard Full Page

Profile5: Standard Full Page

Profile6: Standard Full Page

Profile7: Standard Full Page

Profile8: Standard Full Page

Profile9: Standard Full Page

Profile10: Standard Full Page

Profile11: Standard Full Page

Profile12: Standard Full Page

Profile13: Standard Full Page

Profile14: Standard Full Page

**Figure 2:** Cover Page Setup

---

## **FaxPack.AddRecipient**

### **Summary:**

Add a recipient to a message.

### **Purpose:**

The FaxPack.AddRecipient method is used to add a recipient to a message. A message must have at least one recipient, but there is no limit to the number of additional recipients a message may have. This command provides access to all of the fields available within the 4-Sight FAX address book, except with this plug-in command, all of the field information will come directly from the REALbasic database.



**IMPORTANT:** You must call the FaxPack.AddRecipient command at least once, after calling FaxPack.StartMessage and before calling FaxPack.SendMessage.

### **Syntax:**

**FaxPack.AddRecipient**(fullName, company, destinationKind, faxEmail, countryCode, areaCode, otherFields)

Part	Type	Description
FullName	String	A string expression providing the full name of the recipient of this transmission. This information is not used in the actual computer addressing.
Company	String	A string expression providing the Company (or postal address) of the recipient, used on the cover page of a fax.
Destination Kind	UInt8	A numeric value specifying 1 of three alternatives: 1 = Fax, 2 = Fax with Smart Dial and 3 = Email.
FaxEmail	String	A string expression providing the Fax Number or Email Address depending upon the value of DestKind. This is the field used by Fax Server to transmit the message.
CountryCode	String	A string expression used if "DestKind" is 2 for SMART Dialing. The value is the dialing string required to access that country by telephone.
AreaCode.	String	Dialing string required for recipients area code. Only used if "DestKind" is 1 (for Fax with SMART Dialing).
Other fields	String Array	<i>Array of string values where the index values are as follows: PLEASE SEE BELOW</i>
Salutation	String Array Index value = 0	Contact's salutation.
First Name	String Array Index value = 1	Contact's first name.

Part	Type	Description
Last Name	String Array Index value = 2	Contact's last name.
Title	String Array Index value = 3	Contact's title.
Category	String Array Index value = 4	Contact's category. Examples are employee, vendor, contractor, etc.
Alternate	String Array Index value = 5	Contact's alternate phone number.
Voice	String Array Index value = 6	Contact's phone number.
Email	String Array Index value = 7	Contact's email address.
Address1	String Array Index value = 8	Contact's Address 1.
Address2	String Array Index value = 9	Contact's Address 2.
City	String Array Index value = 10	Contact's City.
State/Province	String Array Index value = 11	State or province.
Country	String Array Index value = 12	Contact's country.
Postal Code	String Array Index value = 13	Contact's postal code.
Custom 1	String Array Index value = 14	Custom 1 field.
Custom 2	String Array Index value = 15	Custom 2 field.
Custom 3	String Array Index value = 16	Custom 3 field.

**Example:**

*Dim otherFields() as String*

*otherFields=Array("Mr.", "John", "Adams", "President", "Customer", "770-454-9401",  
"770-454-9400", "sales@adamsco.com", "123 Main Street", "Suite 200", "Atlanta", "GA",  
"USA", "30345", "blue", "ferrari", "2007")*

*FaxPack.AddRecipient("John Adams", "Adams Company", 0, "770-454-9800", "", "",  
otherFields)*

### Smart Dialing Setup on 4-Sight FAX Server:

The two FAX Server Smart dial dialogs must be configured in order for SMART dialing to be relevant to FaxPack Users.

**Dialing Setup**

Country/Region:

Country code:

Area code:

International dial code:

Long distance dial code:

Smart dial  
 Always dial long distance  
 Always dial International

**Figure 3:** Smart Dialing Setup on 4-Sight FAX Server

The Smartdial feature must be configured specifically for your calling area. Smartdial capabilities exist for all Users.

**Area Code Rules**

Always dial area code (10-digit dialing)

Outside my area code:  
Do not dial 1 for the following area codes:

Within my area code:  
Dial 1 for the following prefixes:

**Figure 4:** Area Code Rules configuration

---

## FaxPack.AddEnclosure

### Summary:

Add an enclosure to a message

### Purpose:

The FaxPack.AddEnclosure method is used to add a TIFF file as an attachment to a message. The TIFF file is usually one that has been previously created by the 4-Sight Fax printer.



**IMPORTANT:** The TIFF file must have certain properties in order to be compatible with 4-Sight Fax, therefore, the 4-Sight FAX print driver must first create it.

### Syntax:

**FaxPack.AddEnclosure(file)**

Part	Type	Description
File	FolderItem	A Tiff File. Usually one that was created with the 4-Sight FAX print driver.

### Example:

```
Dim f as FolderItem
F=GetOpenFolderItem("EnclosureTypes.AllTypes")
If(f <> Nil then)
    FaxPack.AddEnclosure(f)
End If
```

---

## FaxPack.AddComment

### Summary:

Add a comment to a message

### Purpose:

The FaxPack.AddComment command can be used to specify a text comment for a message's cover page. You should call this command only after calling FaxPack.StartMessage and before calling FaxPack.SendMessage. You may only specify one cover-page comment per message.



**IMPORTANT:** Issuing the FaxPack.AddComment command will force a cover page to be printed even when a cover page number is not explicitly referenced in the "Header Page" parameter of the FaxPack.StartMessage command. 4-Sight FAX stores and utilizes a default cover page stored in a resource on the 4-Sight FAX Server.

**Syntax:**

```
FaxPack.AddComment(comment)
```

Part	Type	Description
Comment	String	The comment text to be displayed in the comments section on the header page of a fax or in the body of an email.

**Example:**

```
FaxPack.AddComment("This text will appear in the comments section of the cover page.")
```

**FaxPack.SendMessage****Summary:**

Finalizes the current message and sends it to the 4-Sight FAX Server.

**Purpose:**

The FaxPack.SendMessage command is used to finalize the current message and send it to the 4-Sight FAX Server. The accumulated information, addresses and enclosures specified by the previous calls to the other FaxPack commands are written to disk and sent to the currently specified server. A Message ID is assigned to a specific outgoing fax by the 4-Sight FAX v7 FAX server. FaxPack.SendMessage returns the Message ID and it can be later referenced using the FaxPack.GetItemStatus call to get the current status of that message.

**Syntax:**

```
result=FaxPack.SendMessage
```

Part	Type	Description
Result	UInt32	Message ID returned by the 4-Sight FAX Server. Returns zero if an error occurred.

**Example:**

```
Dim messageID as UInt32  
messageID=FaxPack.SendMessage
```

---

## FaxPack.AbandonMessage

### Summary:

Abandons the current message and disposes of its allocated resources.

### Purpose:

Abandons the current message and disposes of all its allocated resources. It is typically called if any of the message building methods fails, or if the application has to abandon the message for any reason.

### Syntax:

**FaxPack.AbandonMessage**

### Example:

*FaxPack.AbandonMessage*

---

## FaxPack.GetItemStatus

### Summary:

Retrieves the status, state and other related information for a particular message (fax or email) from the 4-Sight FAX Server.

### Purpose:

The FaxPack.GetItemStatus command retrieves the status, state and other related information for a particular fax or email message. This command is often used in conjunction with (and subsequent to) the FaxPack.GetQueueListing command, which allows developers to obtain the Unique ID for any fax. *For a complete description of the Item Status Class, please refer to the Item Status sub-section within the Classes section.*



**IMPORTANT:** The idKind (2<sup>nd</sup>) parameter is evaluated first to determine the nature of the value within the “id” parameter.

### Syntax:

result=**FaxPack.GetItemStatus**(idKind, id, index)

Part	Type	Description
result	ItemStatus	A FaxPack ItemStatus class instance. Returns Nil if an error occurred.
idKind	UInt8	0 = Message ID, 1 = Outgoing Queue Item ID, 2 = Incoming Queue Item ID.

Part	Type	Description
id	UInt32	Either a Message ID, Outgoing Queue Item ID, or an Incoming Queue ID depending upon the 'idKind' parameter.
index	UInt32	The index of the recipient if the 'idKind' parameter is zero. The first recipient is an index value of 1.

### Example

```

Dim messageID as UInt32
messageID=FaxPack.SendMessage() // send our current message to the 4-Sight FAX
Server
if messageID <> 0 then
    dim messageInfo as ItemStatus
    messageInfo=FaxPack.GetItemStatus(0, messageID, 1)
    if messageInfo <> Nil then
        MsgBox("The status of the fax for "+messageInfo.mTo+" is "+messageInfo.mStatus)
    end if
end if

```

## Printer Control

---

### FaxPack.SwitchPrinter

#### Summary:

Changes the default printer to the 4-Sight FAX printer (driver).

#### Purpose:

The FaxPack.SwitchPrinter changes the default printer to/from the 4-Sight FAX printer. The current default printer is stored before changing to the 4-Sight FAX printer, and then restored when changing back.

As an example, if the user currently has a Hewlett Packard (HP) printer device file as the default printer, this command can be used to explicitly set 4-Sight FAX as the current printer, or subsequently revert to the HP as the current printer, or to toggle between the two.

#### Syntax:

**FaxPack.SwitchPrinter**(Mode)

Part	Type	Description
Mode	UInt8	1 = restore default printer, 2 = switch to the 4-Sight FAX print driver.

#### Example:

```
FaxPack.SwitchPrinter(2) // change default printer to the 4-Sight FAX printer  
FaxPack.SwitchPrinter(1) // change default printer back
```

---

### FaxPack.PreparePrint

#### Summary:

Used to prepare for capturing the subsequent print job as an attachment to the current message.

#### Purpose:

The FaxPack.PreparePrint method is used to prepare for capturing the subsequent print job as an attachment to the current message. This call must be used before each and every print job that is to be added as an attachment. Printing must be done through the 4-Sight FAX printer.

**Syntax:**

**FaxPack.PreparePrint**(resolution, grayscale)

Part	Type	Description
Resolution	Uint8	1 = standard resolution, 2 = fine resolution
grayScale	Uint8	0 = No grayscale, 1 = grayscale

**Example:**

```
FaxPack.SwitchPrinter(2) // switch to 4-Sight FAX printer  
FaxPack.PreparePrint(2,1) // prepare capturing of the subsequent print job  
Dim page as Graphics  
page=OpenPrinter()  
if page <> Nil then  
    page.DrawString("Hello User",100,50)  
End if  
FaxPack.SwitchPrinter(1) // switch default printer back
```

## Server Control

---

### FaxPack.ServerLookup

#### Summary:

Searches for 4-Sight FAX-Servers within your Local Area Network.

#### Purpose:

The FaxPack.ServerLookup command allows you to retrieve the list of 4-Sight FAX-Servers found in your Local Area Network (LAN). This step is valuable when the application is first installed or the 4-Sight FAX server configuration is changed. After the user selects a server, future messages are sent to the selected server when FaxPack.SendMessage is called.

The command will return a description of the chosen server name, the associated IP address and the Server port number for each 4-Sight FAX server found. To select a server, the values should be passed to the FaxPack.ServerLogin command for subsequent sessions. Within each TCP/IP network, it may take several seconds to search, locate and return all of the local area network fax servers. The default time for fax server lookup and display is 5 seconds. If more time is needed because of either a larger or slower network, the REALbasic Developer can increase the "TMEOUT" time to a larger number.

CAUTION: REALbasic Developers are not encouraged to use large numbers in the TimeOut parameter, because the command may wait until this number of seconds has transpired before returning values. The MaxToFind parameter, usually a value of 1, allows you to specify how many 4-Sight Servers to search for. The search will stop either when the MaxToFind or Timeout has been reached (whichever happens first). *For a complete description of the Server List class, please refer to the ServerList sub-section within the Classes section.*

#### Syntax:

```
result=FaxPack.ServerLookup(TimeOutSecond, MaxToFind)
```

Part	Type	Description
Result	ServerList	A FaxPack ServerList class instance. Returns Nil if an error occurred.
timeOutSeconds	UInt16	Maximum number of seconds to do a search.
maxToFind	UInt16	Maximum number of 4-Sight FAX Servers to search for.

#### Return Value:

Name	Explanation
ResultCode	If the operation is successful, a value of 0 is returned. If an error occurs, a non-zero error code value is returned.

**Example:**

```

Dim fxServers as ServerList
fxServers=FaxPack.ServerLookup(10, 1)
if fxServers <> Nil then
    if fxServers.Count > 0 then
        Dim faxServerItem as ServerItem
        faxServerItem=fxServers.Item(0)
        MsgBox("Found server "+faxServerItem.mName+"@"+faxServerItem.mIPAddress)
    end if
end if

```

**FaxPack.ServerLogin****Summary:**

Establishes the 4-Sight FAX Server and User to be used by subsequent FaxPack commands.

**Purpose:**

The FaxPack.ServerLogin command is used to request FAX server access or to change the current FAX server for pending and future messages in the current application session. The server description is obtained by calling the FaxPack.ServerLookup commands.

If the specified server is valid and available, and the Username and password are also valid, then the connection is established and messages will be sent to the specified server when the FaxPack.SendMessage command is called.

If the specified server is not available and the values passed are incorrect, the REALbasic developer should reissue the FaxPack.ServerLookup command to obtain a list of valid servers. After a valid server is logged into, messages will be sent to the selected server when the FaxPack.SendMessage command is called. Messages cannot be sent to a server until the server login process has been completed.

**Syntax:**

**FaxPack.ServerLogin**(ipAddress, ipPort, userName, password, trialLogin)

Part	Type	Description
ipAddress	String	TCP/IP address of the 4-Sight FAX Server.
ipPort	UInt16	TCP/IP port of the 4-Sight FAX Server. (0 = use default)
userName	String	Username from list of 4-Sight FAX Server Users.
password	String	User's password.
trialLogin	UInt8	(0 = no, 1 = yes)

**Example:**

```
FaxPack.ServerLogin("192.168.0.1", 0, "Jason Parsley", "abc", 0)
```

**FaxPack.GetCoverPages****Summary:**

*Retrieves a list of cover page names from the current 4-Sight FAX Server.*

**Purpose:**

The FaxPack.GetCoverPages command allows you to retrieve a list of the cover page names defined in the current server. *For a complete description of the CoverPageList class, please refer to the CoverPageList sub-section within the Classes section.*

**Syntax:**

```
result=FaxPack.GetCoverPages
```

Part	Type	Description
Result	CoverPageList	A FaxPack CoverPageList class instance. Returns Nil if an error occurred.

**Example:**

```
Dim fxCoverPages as CoverPageList
fxCoverPages=FaxPack.GetCoverPages
if fxCoverPages <> Nil then
    if fxCoverPages.Count > 0 then
        MsgBox("First cover page is "+fxCoverPages.Item(0))
    End If
End If
```

**FaxPack.GetQueueListing****Summary:**

Downloads a queue listing from the current 4-Sight FAX Server to a local tab delimited file.

**Purpose:**

The FaxPack.GetQueueListing command downloads a queue listing from the FAX Server into a local tab delimited text file saved on the client machine issuing this call. The queue file provides an equivalent row along with all its fields for each fax record within the queue specified in the "Kind" parameter. All of the values within a specific row comprises all of the data available for each fax. Command descriptions and file details are as follows.

**Syntax:**

result=**FaxPack.GetQueueListing**(file, listKind)

Part	Type	Description
result	UInt32	Number of items in the listing including the column header. Returns zero if an error occurred or if the listing is empty.
file	FolderItem	Destination file that is to contain the listing.
listKind	UInt16	(1 = all faxes out, 2 = all faxes in, 3 = my faxes out, 4 = my faxes in)

**Text file format for incoming queue listing:**

Name	Explanation
ID	A numeric value containing the Fax message ID number.
Date	A date value containing the date of the fax.
To	A string value containing the recipient name for the facsimile.
Subject	A string value containing the subject for the facsimile.
State	A value containing the state of the facsimile. Because all values within this text file are essentially string, this value should be converted within the REALbasic program to a numeric value. The State contains flags where Hex 0x01= "sent", read or printed", where 0x02= "suspended", where "0x04" = "sent", where "0x08" = "rejected" and where "0x10" = "maximum retries".
Status	A string value containing the current status of the facsimile where the following possible values exist as of Version 7.0: "Preparing page: #/#", "Page: #/#", "Sending", "<Suspended>", "Failed", "Waiting", "Deleting", "Deleted", "Sent", "Failed", "Printing", "Archiving", "Max Retries", "Stopping...", "Printed", "Rejected", "Imaging Page", "Prep: # of #", "Converting", "Received", "Unconverted", "Imaged", "Read", "Print", "Printing", "Preparing # of #", "Sending", "Connecting...", "Connected", "Sending Page: #", "Receiving Page #". This list is subject to change, so it is not advisable to build programmer logic around the values.
Pages	A value containing the number of pages for the facsimile. This string should be converted within the REALbasic program to a numeric value.
Size	A value containing the physical size, in bytes, of the facsimile for the facsimile. Within the text file, this is stored as a string and should be converted within the REALbasic program to a numeric value.
From	A string value containing the person who sent the facsimile.
FromNumber	A string value containing the fax number of the person who sent the facsimile.

## Text file format for outgoing queue listing:

Name	Explanation
ID	A numeric value containing the Fax message ID number.
Date	A Date value containing the date of the fax.
From	A string value containing the person's name who sent the facsimile.
To	A string value containing the recipient's name for the facsimile.
Subject	A string value containing the subject for the facsimile.
State	A value containing the state of the facsimile. Because all values within this text file are essentially string, this value should be converted within the REALbasic program to a numeric value. The State contains flags where Hex 0x01= "sent", read or printed", where 0x02= "suspended", where "0x04" = "sent", where "0x08" = "rejected" and where "0x10" = "maximum retries".
Status	A string value containing the current status of the facsimile where the following possible values exist as of 4-Sight FAX Version 7.0: "Preparing page: #/#", "Page: #/#", "Sending", "<Suspended>", "Failed", "Waiting", "Deleting", "Deleted", "Sent", "Failed", "Printing", "Archiving", "Max Retries", "Stopping...", "Printed", "Rejected", "Imaging Page", "Prep: # of #", "Converting", "Received", "Unconverted", "Imaged", "Read", "Print", "Printing", "Preparing # of #", "Sending", "Connecting...", "Connected", "Sending Page: #", "Receiving Page #". This list is subject to change, so it is not advisable to build programmer logic around the values.
Position	A numeric value containing the queue item position.
Priority	A string value indicating the priority for this fax. Values range from 1 (lowest) to 5 (highest).
Reason	A string value containing the reason the fax was rejected.
CommType	An numeric value where 1 = fax and 2 = email. Within the text file, this is stored as a string and should subsequently be converted in the REALbasic program to a numeric value.

### Example:

```
Dim file as FolderItem
file=GetSaveFolderItem("", "QueueListing")
if file <> Nil then
    numItems=FaxPack.GetQueueListing(file, 3)
    if numItems > 0 then
        // open the file and parse the listing (the first line is the column headers)
    End If
    File.Delete()
End if
```

---

## **FaxPack.GetFaxTiff**

### **Summary:**

Downloads a TIFF file of a given queue item from the current 4-Sight FAX Server.

### **Purpose:**

The FaxPack.GetFaxTIFF downloads a TIFF file of a given queue item from the current 4-Sight FAX Server. This command is useful for Tiff File retrieval for databases providing facsimile viewing capabilities or local archival. Developers can either retrieve: 1) a one page preview (Part = 1) (for the first page) 2) the cover page (Part = 2) or 3) the attachments for outgoing faxes (Part = 3).

### **Syntax:**

**FaxPack.GetFaxTiff**(WhichQueue, ItemID, whichPart, Attachment Number, File)

Part	Type	Description
WhichQueue	UInt16	(1 = outgoing queue, 2 = incoming queue)
ItemID	UInt32	Queue Item ID
WhichPart	UInt8	(1 = one-page Preview, 2 = cover page, outgoing fax only, 3 = attachment.
AttachmentNumber	UInt16	Attachment number if the 'whichPart' parameter is 3.
File	FolderItem	Destination file which is to contain the TIFF.

### **Example:**

```
Dim file as FolderItem
// itemID is was obtained from the queue listing
file=GetSaveFolderItem("", "QueueListing")
if file <> Nil then
    FaxPack.GetFaxTIFF(1, itemID, 3, 1, file)
end if
```

## Image Control

---

### FaxPack.GetTiffInfo

#### Summary:

Returns the number of pages in the given TIFF file.

#### Purpose

The FaxPack.GetTiffInfo command returns the number of pages in the Tiff file specified. Retrieve the Tiff file from the FAX Server first using the FaxPack.GetFaxTiff command because the FaxPack.GetTiffInfo command operates on files from the local hard drive.

#### Syntax:

```
result=FaxPack.GetTIFFInfo(file)
```

Part	Type	Description
Tiff File Path	UInt16	Number of pages in the TIFF file. Returns zero if an error occurred.
file	FolderItem	TIFF file.

#### Example:

```
Dim numPages as UInt16  
Dim tiffFile as FolderItem  
tiffFile=GetOpenFolderItem(PictureTypes.TIFFTypes)  
if tiffFile <> Nil then  
    numPages=FaxPack.GetTIFFInfo(tiffFile)  
    MsgBox("Number of pages in the TIFF file is "+Str(numPages))  
End If
```

---

### FaxPack.ConvertTiff

#### Summary:

Converts a given TIFF file to another file format.

#### Purpose:

The FaxPack.ConvertTiff command converts the Fax Tiff file into any of 5 different file formats including PDF, Pict, PNG, BMP and Tiff. Developers can optionally specify the page range for each file. Additionally, the developer must specify the Tiff source file full path name as well as the destination file path.



**IMPORTANT:** Prior to calling the FaxPack.ConvertTiff command, the developer should call the FaxPack.GetFaxTiff command to first retrieve the Tiff file to be converted.

**Syntax:**

**FaxPack.ConvertTIFF**(convertType, srcFile, dstFile, startPage, endPage)

Part	Type	Description
convertType	UInt8	A value specifying the file type, where 1=PDF, 2=Pict, 3 = PNG, 4 = BMP and 5 = Tiff.
srcFile	FolderItem	Source TIFF file.
dstFile	FolderItem	A FolderItem pointing to where the converted image is to be saved.
startPage	UInt16	Page number of the first page to convert.
endPage	UInt16	Page number of the last page to convert.

**Example:**

```

Dim numPages as UInt16
Dim tiffFile as FolderItem
tiffFile=GetOpenFolderItem(PictureTypes.TIFFTypes)
if tiffFile <> Nil then
    dstFile=GetSaveFolderItem("", "FirstPage.pdf")
    if dstFile <> Nil then
        numPages=FaxPack.GetTIFFInfo(tiffFile)
        FaxPack.ConvertTIFF (1, tiffFile, dstFile, 1, 1) // convert page1 of TIFF file to PDF
    end if
end if
end If

```

## Address Book

### Address Book Privileges:

Please contact your 4-Sight FAX Administrator to ensure those FaxPack Users who intend to work with the 4-Sight FAX Address Book are set up to do so. The following screen shot from the User setup dialog shows the boxes to be checked to enable all Address Book privileges from a FaxPack application.

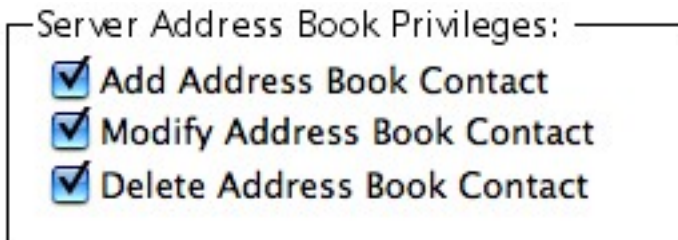


Figure 5: Address Book Privileges

---

## FaxPack.GetAddressBook

### Summary:

Downloads the entire address book from the current 4-Sight FAX Server.

### Purpose:

The FaxPack.GetAddressBook method downloads the entire address book from the current 4-Sight FAX Server.

### Syntax:

**FaxPack.GetAddressBook**(file, encodingKind)

Part	Type	Description
file	FolderItem	Destination file to contain the address book.
encodingKind	UInt16	0 = MacRoman, 1 = UTF8

### Example:

```
Dim file as FolderItem
file=GetSaveFolderItem("", "Server Address Book")
if file <> Nil then
    FaxPack.GetAddressBook(file, 1)
    // open the file and parse the listing (the first line is the column headers)
end if
```

---

## **FaxPack.GetContact**

### **Summary:**

Retrieves contact information with the given ID from the current 4-Sight FAX Server.

### **Purpose:**

The FaxPack.GetContact function retrieves contact information with the given ID from the current 4-Sight FAX Server. *For a complete description of the ContactItem class, please refer to the ContactItem sub-section within the Classes section.*

### **Syntax:**

result=**FaxPack.GetContact**(id)

Part	Type	Description
result	ContactItem	A FaxPack ContactItem class instance. Returns Nil if an error occurred.
id	UInt32	A contact ID.

### **Example:**

```
Dim fxContactItem As ContactItem
// contactID was obtained from the server address book listing
fxContactItem=FaxPack.GetContact(contactID)
if fxContactItem <> Nil then
    MsgBox(fxContactItem.GetField(ContactItem.kFULL_NAME) + " is from "
+fxContactItem.GetField(ContactItem.kCOMPANY))
end if
```

---

## **FaxPack.ModifyContact**

### **Summary:**

Modifies contact information with the given ID at the current 4-Sight FAX Server.

### **Purpose:**

The FaxPack.ModifyContact method modifies contact information with the given ID at the current 4-Sight FAX Server. *For a complete description of the ContactItem class, please refer to the ContactItem sub-section within the Classes section.*

### **Syntax:**

**FaxPack.ModifyContact**(id, contact)

Part	Type	Description
id	UInt32	A contact ID.
contact	ContactItem	A FaxPack ContactItem class instance.

**Example:**

```

Dim fxContactItem As ContactItem
// contactID was obtained from the server address book listing
fxContactItem=FaxPack.GetContact(contactID)
if fxContactItem <> Nil then
    fxContactItem.SetField(ContactItem.kTITLE, "Senior Partner") // change the contact's title
    FaxPack.ModifyContact(contactID, fxContactItem)
end if

```

**FaxPack.AddContact**

**Summary:**

Adds a contact to the list at the current 4-Sight FAX Server.

**Purpose:**

The FaxPack.AddContact adds a contact to the list at the current 4-Sight FAX Server. *For a complete description of the ContactItem class, please refer to the ContactItem subsection within the Classes section.*

**Syntax:**

```

result=FaxPack.AddContact(contact)

```

Part	Type	Description
result	UInt32	A contact ID.
contact	ContactItem	A FaxPack ContactItem class instance.

**Example:**

```

Dim fxContactItem As ContactItem
Dim contactID As UInt32
fxContactItem = new ContactItem
fxContactItem.SetField(ContactItem.kFULL_NAME, "Mickey Mouse") // Full Name
fxContactItem.SetField(ContactItem.kCOMPANY, "Walt Disney") // Company
fxContactItem.SetField(ContactItem.kNUMBER, "1-800-555-1212") // Fax Number
fxContactItem.SetField(ContactItem.kCITY, "Orlando") // City
fxContactItem.SetField(ContactItem.kSTATE, "Florida") // State

```

```

fxContactItem.SetField(ContactItem.kCOUNTRY, "USA") // Country
contactID=FaxPack.AddContact(fxContactItem)
if contactID <> 0 then
    MsgBox("Contact added; its ID is "+Str(contactID))
end if

```

---

## **FaxPack.DeleteContact**

### **Summary:**

Deletes a contact from the list at the current 4-Sight FAX Server.

### **Purpose:**

The FaxPack.DeleteContact method deletes a contact from the list at the current 4-Sight FAX Server. *For a complete description of the ContactItem class, please refer to the ContactItem sub-section within the Classes section.*

### **Syntax:**

**FaxPack.DeleteContact(id)**

Part	Type	Description
id	UInt32	A contact ID.

### **Example:**

```

// contactID was obtained from the server address book listing
FaxPack.DeleteContact(contactID)

```

## Block List

### FaxPack.AddToBlockList

#### Summary:

Adds the given name & phone number to the blocked list at the current 4-Sight FAX Server.

#### Purpose:

The FaxPack.AddToBlockList method adds the given name & phone number to the blocked list at the current 4-Sight FAX Server.

#### Syntax:

**FaxPack.AddToBlockList**(name, phoneNumber)

Part	Type	Description
name	String	The name associated with the phone number.
phoneNumber	String	The phone number to add to the blocked list.

#### Example:

*FaxPack.AddToBlockList("Junk Faxes", "800-123-4567")*

#### Login and Access:



The 4-Sight FAX Block List capabilities must be enabled from the 4-Sight FAX Server for each User who will be using this feature. To allow a User to add to the Fax Block list, check the "Allow add to Block List". To allow a User to delete to the Fax Block list, check the "Allow delete from Block List". For example, please review the following FAX server screen:

Login and Access:

- Administrator access to server
- Full queue access
- Allow automatic login
- Allow username alias
- Enforce group name as public
- Allow add to Block List
- Allow delete from Block List

**Figure 6:** Login and Access

---

## **FaxPack.DeleteFromBlockList**

### **Summary:**

Deletes the phone number from the blocked list at the current 4-Sight FAX Server.

### **Purpose:**

The FaxPack.DeleteFromBlockList command deletes the fax number from the 4-Sight FAX Server Block List including its associated name. By doing this, future incoming faxes from this fax number will no longer be halted by the 4-Sight FAX Server.

### **Syntax:**

**FaxPack.DeleteFromBlockList(phoneNumber)**

Part	Type	Description
phoneNumber	String	The phone number to delete from the blocked list.

### **Example:**

*FaxPack.DeleteFromBlockList("800-123-4567")*

# Command Summary (Methods and Functions)

The following table is a summary of all of the FaxPack commands and their syntax.

Command	Syntax
FaxPack.Version	result= <b>FaxPack.Version</b>
FaxPack.StartMessage	<b>FaxPack.StartMessage</b> (From, Subject, CoverPage, Priority, SendDateTime)
FaxPack.AddRecipient	<b>FaxPack.AddRecipient</b> (fullName, company, destinationKind, faxEmail, countryCode, areaCode, otherFields)
FaxPack.AddEnclosure	<b>FaxPack.AddEnclosure</b> (file)
FaxPack.AddComment	<b>FaxPack.AddComment</b> (comment)
FaxPack.SendMessage	result= <b>FaxPack.SendMessage</b>
FaxPack.AbandonMessage	<b>FaxPack.AbandonMessage</b>
FaxPack.GetItemStatus	result= <b>FaxPack.SendMessage</b>
FaxPack.SwitchPrinter	<b>FaxPack.SwitchPrinter</b> (Mode)
FaxPack.PreparePrint	<b>FaxPack.PreparePrint</b> (resolution, grayscale)
FaxPack.ServerLookup	result= <b>FaxPack.ServerLookup</b> (TimeOutSecond,MaxToFind)
FaxPack.ServerLogin	<b>FaxPack.ServerLogin</b> (ipAddress, ipPort, userName, password, trialLogin)
FaxPack.GetCoverPages	result= <b>FaxPack.GetCoverPages</b> ()
FaxPack.GetQueueListing	result= <b>FaxPack.GetQueueListing</b> (file, listKind)
FaxPack.GetFaxTiff	<b>FaxPack.GetFaxTiff</b> (WhichQueue, ItemID, whichPart, Attachment Number, File)
FaxPack.GetTiffInfo	result= <b>FaxPack.GetTIFFInfo</b> (file)
FaxPack.ConvertTiff	<b>FaxPack.ConvertTiff</b> (convertType, srcFile, dstFile, startPage, endPage)
FaxPack.GetAddressBook	<b>FaxPack.GetAddressBook</b> (file, encodingKind)
FaxPack.GetContact	result= <b>FaxPack.GetContact</b> (id)
FaxPack.ModifyContact	<b>FaxPack.ModifyContact</b> (id, contact)
FaxPack.AddContact	result= <b>FaxPack.AddContact</b> (contact)
FaxPack.DeleteContact	<b>FaxPack.DeleteContact</b> (id)
FaxPack.AddToBlockList	<b>FaxPack.AddToBlockList</b> (name, phoneNumber)
FaxPack.DeleteFromBlockList	<b>FaxPack.DeleteFromBlockList</b> (phoneNumber)

# Properties

## *FaxPack Properties*

### **Description:**

The FaxPack module has two properties: "ErrorCode" & "RaiseOption".

The "ErrorCode" property contains the error value of the last FaxPack method or function called. A value of zero means the call was successful.

The "RaiseOption" property provides you with an option to use an Exception block or Try block on FaxPack method or function calls.

Set the RaiseOption to a value of one if you want FaxPack methods or functions to raise an exception when an error occurs. Set the RaiseOption to a value of zero if you do not want exceptions raised.

### **Properties:**

Name	Type	Description
ErrorCode	Int16	Error code value of the last FaxPack method or function that was called.
RaiseOption	UInt8	Whether or not a FaxPack method or function should raise an exception. (0 = no, 1 = yes)

### **Example:**

```
// An Example using FaxPack.ErrorCode  
FaxPack.AddComment("Here are some comments.")  
if FaxPack.ErrorCode=0 then  
    MsgBox("FaxPack.AddComment was successful")  
else  
    MsgBox("FaxPack.AddComment failed, error#" + Str(FaxPack.ErrorCode))  
end if
```

*For example:*

```
// An Example using FaxPack.RaiseOption=1  
FaxPack.RaiseOption=1  
try  
    FaxPack.ServerLogin("127.0.0.1", 0, "Mickey", "Disney", 1)  
    MsgBox("The Fax Server login was successful!")  
catch err  
    MsgBox("Fax Server login failed, error#", Str(err.ErrorNumber))  
end try
```

# Constants

## Important Note:

Developers: Please ensure you insert the “FAXPACK.” Prefix before each Class Constants shown within the tables below. For example, in lieu of “kPRIORITY\_BULK”, you would actually type “FAXPACK.kPRIORITY\_BULK”.

## Priority

For “priority” parameter in FaxPack.StartMessage

Class Constant	Value
kPRIORITY_BULK	1
kPRIORITY_NORMAL	2
kPRIORITY_IMPORTANT	3
kPRIORITY_ASAP	4
kPRIORITY_URGENT	5

## DestinationKind

For “destinationKind” parameter in FaxPack.StartMessage

Class Constant	Value
kDESTINATION_FAX	1
kDESTINATION_FAX_SMARTDIAL	2
kDESTINATION_EMAIL	3

## otherFields

Index values for “otherFields” array in FaxPack.AddRecipient

Class Constant	Value
kRECIPIENT_SALUTATION	0
kRECIPIENT_FIRST_NAME	1
kRECIPIENT_LAST_NAME	2
kRECIPIENT_TITLE	3

Class Constant	Value
kRECIPIENT_CATEGORY	4
kRECIPIENT_ALTERNATE	5
kRECIPIENT_VOICE	6
kRECIPIENT_EMAIL	7
kRECIPIENT_ADDRESS1	8
kRECIPIENT_ADDRESS2	9
kRECIPIENT_CITY	10
kRECIPIENT_STATE	11
kRECIPIENT_COUNTRY	12
kRECIPIENT_POSTAL_CODE	13
kRECIPIENT_CUSTOM1	14
kRECIPIENT_CUSTOM2	15
kRECIPIENT_CUSTOM3	16

## ***Resolution***

For “Resolution” parameter in FaxPack.PreparePrint

Class Constant	Value
kRESOLUTION_STANDARD	1
kRESOLUTION_FINE	2

## ***convertType***

For “convertType” parameter in FaxPack.ConvertTiff

Class Constant	Value
kCONVERT_TYPE_PDF	1
kCONVERT_TYPE_PICT	2
kCONVERT_TYPE_PNG	3
kCONVERT_TYPE_BMP	4
kCONVERT_TYPE_TIFF	5

# Classes

## CoverPageList

### Description:

Holds a list of cover page names.

The CoverPageList class holds a list of cover pages names. A CoverPageList instance is obtained with a call to the FaxPack.GetCoverPages function. The number of items in the list can be obtained with a call to the Count function. Pass an index, zero-based, to the Item function to get one of the cover page names.

### Super Class:

Object.

### Methods:

Name	Parameters	Return Type	Description
Count		UInt16	Returns the number of items in the list.
Item	index as UInt16	String	Returns a cover page name with the given index. The first item is an index value of 0.

### Examples:

```
Dim fxCoverPages as CoverPageList
fxCoverPages=FaxPack.GetCoverPages
if fxCoverPages <> Nil then
    // getting the number of cover pages
    MsgBox("Number of cover pages is "+Str(fxCoverPages.Count()))
    if fxCoverPages.Count > 0 then
        // getting one of the cover page names
        MsgBox("First cover page is "+fxCoverPages.Item(0))
    end if
end if
```

## ServerItem

### Description:

Holds the name and TCP/IP information of a 4-Sight FAX Server.

The ServerItem holds the name and TCP/IP information of a 4-Sight FAX Server. A ServerItem instance can be obtained from the Item function of a ServerList instance.

**Super Class:**

Object.

**Properties:**

Name	Type	Description
mName	String	Name of the machine running a 4-Sight FAX Server.
mIPAddress	String	TCP/IP address of a 4-Sight FAX Server.
mPort	UInt16	TCP/IP port of a 4-Sight FAX Server.

**Example:**

```

Dim fxServers as ServerList
fxServers=FaxPack.ServerLookup(10, 1)
if fxServers <> Nil then
    if fxServers.Count > 0 then
        Dim fxServerItem as ServerItem
        fxServerItem=fxServers.Item(1)
        MsgBox("Found server "+fxServerItem.mName+"@"+fxServerItem.mIPAddress+"."
+Str(fxServerItem.mPort))
    end if
end if

```

## ServerList

**Description :**

Holds a list of ServerItem instances.

The ServerList class holds a list of ServerItem instances. A ServerList instance can be obtained with a call to the FaxPack.ServerLookup function. The number of items in the list can be obtained with a call to the Count function. Pass an index, zero-based, to the Item function to get one of the ServerItem instances.

**Super Class:**

Object.

**Methods:**

Name	Parameters	Return Type	Description
Count		UInt16	Returns the number of items in the list.
Item	index as UInt16	ServerItem	Returns a ServerItem instance with the given index. The first item is an index value of 0.

**Example:**

```

Dim fxServers as ServerList
fxServers=FaxPack.ServerLookup(10, 1)
if fxServers <> Nil then
    if fxServers.Count > 0 then
        Dim fxServerItem as ServerItem
        fxServerItem=fxServers.Item(0)
        MsgBox("Found server "+fxServerItem.mName+"@"+fxServerItem.mIPAddress+"."
+Str(fxServerItem.mPort))
    end if
end if

```

## ItemStatus

### Description:

Holds the status, state and other related information for a particular fax or email message.

The ItemStatus class holds the status, state and other related information for a particular fax or email message. An ItemStatus instance can be obtained from a call to the FaxPack.GetItemStatus function.

### Super Class:

Object.

### Properties:

Name	Type	Description
mFrom	String	Whom the message is from.
mTo	String	Whom the message is to.
mSubject	String	Message subject.
mCommType	UInt8	Message destination kind. (1 = Fax, 2 = Email)
mFaxNumber	String	Fax number if 'mCommType' is 1.
mEmailAddress	String	Email address if 'mCommType' is 2.
mStatus	String	Current status. (e.g. - "Imaged", "Sent", "Received")
mRejectReason	String	Reject reason if 'mRejected' is 1.
mHasCoverPage	UInt8	(0 = no cover page, 1 = has a cover page)
mPageCount	UInt16	Number of total pages in the message.
mAttachmentCount	UInt16	Number of attachments in the message.
mSent	UInt8	(0 = message not sent, 1 = message sent)
mPrinted	UInt8	(0 = message not printed, 1 = message printed)

Name	Type	Description
mDeleted	UInt8	(0 = message not deleted, 1 = message deleted)
mRejected	UInt8	(0 = message not rejected, 1 = message rejected)

**Example:**

```
// a messageID was obtained from a call to FaxPack.SendMessage
dim messageInfo as ItemStatus
messageInfo=FaxPack.GetItemStatus(0, messageID, 1)
if messageInfo <> Nil then
    MsgBox("The status of the fax from "+messageInfo.mFrom+" for
    "+messageInfo.mTo+" is "+messageInfo.mStatus)
    if messageInfo.mSent = 1 then
        MsgBox("The fax has been sent.")
    else
        MsgBox("The fax has not been sent.")
    end if
end if
```

## ContactItem

**Description:**

Holds the information for a contact from the server address book.

The ContactItem class holds the information for a contact from the server address book. A ContactItem instance can be obtained from a call to FaxPack.GetContact.

**Super Class:**

Object.

**Properties:**

Name	Type	Description
mID	UInt32	Contact ID.

**Methods:**

Name	Parameters	Return Type	Description
GetField	index as UInt16	String	Returns a contact field value given a ContactItem field constant index.
SetField	index as UInt16, value as String	Sets a contact field value given a ContactItem field constant index.	

**Class Constants:**

Class Constant	Value
kFULL_NAME	1
kFIRST_NAME	2
kLAST_NAME	3
kSALUTATION	4
kTITLE	5
kCOMPANY	6
kCATEGORY	7
kSMART_DIALING	8
kCOUNTRYCODE	9
kAREACODE	10
kNUMBER	11
kEMAIL_ADDRESS	12
kPHONE_NUMBER	13
kADDRESS_LINE1	14
kADDRESS_LINE2	15
kCITY	16
kSTATE	17
kCOUNTRY	18
kPOSTAL_CODE	19
kCUSTOM1	20
kCUSTOM2	21
kCUSTOM3	22

**Example:**

```
// getting information from a ContactItem
Dim fxContactItem As ContactItem
// contactID was obtained from the server address book listing
fxContactItem=FaxPack.GetContact(contactID)
if fxContactItem <> Nil then
    MsgBox(fxContactItem.GetField(ContactItem.kFULL_NAME) + " is from
"+fxContactItem.GetField(ContactItem.kCOMPANY))
end if

// setting information of a ContactItem
Dim fxContactItem As ContactItem
Dim contactID As UInt32
fxContactItem = new ContactItem
fxContactItem.SetField(ContactItem.kFULL_NAME, "Mickey Mouse") // Full Name
fxContactItem.SetField(ContactItem.kCOMPANY, "Walt Disney") // Company
    // setup to use smart dialing
fxContactItem.SetField(ContactItem.kSMART_DIALING, "1") // 1 = use smart dialing, 0 =
do not use smart dialing
fxContactItem.SetField(ContactItem.kCOUNTRYCODE, "1") // "1" is country code for USA
fxContactItem.SetField(ContactItem.kAREACODE, "800") // area code
fxContactItem.SetField(ContactItem.kNUMBER, "555-1212") // remaining fax number
    // -----
fxContactItem.SetField(ContactItem.kCITY, "Orlando") // City
fxContactItem.SetField(ContactItem.kSTATE, "Florida") // State
fxContactItem.SetField(ContactItem.kCOUNTRY, "USA") // Country
contactID=FaxPack.AddContact(fxContactItem)
if contactID <> 0 then
    MsgBox("Contact added; its ID is "+Str(contactID))
end if
```

# Error Codes

The following table lists possible numeric values returned by the commands described earlier, along with their definitions and the mnemonic names by which they are referred. These values represent errors directly related to the Fax Server software. Unlisted OS codes are possible.

Value	Definition
0	No error; the operation was performed successfully.
-1	The FaxPack demonstration time has expired.
10003	A file could not be opened for reading.
10004	A file could not be opened for writing.
10005	A file was empty.
10006	A read error.
10007	A write error.
10008	Could not obtain a session with the Server.
10009	Message ID was not found.
10010	Fax printer was not found.
10011	A time-out happened when waiting for a Fax printer file to complete.
10052	Invalid User or Password.
10012	4-Sight Fax printer version 5.0.5 or later is required
10013	Must have at least one recipient for an outgoing fax
10014	Printer does not exist
10015	The Fax printer does not support this feature
10023	Invalid Contact ID (usually a value of 0)
10052	Invalid User or Password
10053	User is already logged-in.
10054	Logins are disabled at the Server.
10057	User does not have the permission to perform operation.
10073	"From" name is different than the User name, and the "Allow user name alias" is unchecked at the server.
10090	the Contact with the given ID was not found. New in version 7.
10091	the Contact is being edited by another user. New in version 7.
10108	Communication with the Server timed-out.
10155	Attachment cannot be located.
10185	Attachment or recipient list rejected by the Fax Server.
10200	The current Server has not been set.

Value	Definition
10201	The current login information has not been set.
10202	Print preparation has not been called.
10203	Server is invalid or is not responding.
10204	A memory allocation failed.
10205	Command is unimplemented.
10206	"FaxPack.StartMessage" must be called first.
10207	Message command has been called in the wrong order.
10208	A message is already being created.
10209	A valid 'From' is needed.
10210	Invalid path.
10211	Invalid array type.
10212	Navigation Services are not available.
10213	Must use MacOS X 10.2.2+ for "FaxPack.SwitchPrinter".
10214	There were no calls to "FaxPack.PreparePrint".
10215	There was more than one call to "FaxPack.PreparePrint".
10217	The fax number parameter to "FaxPack.AddRecipient" was empty.
10218	Insufficient number of parameters passed
10219	Could not get parameter
10220	Invalid switch printer code
10221	No servers found
10222	An IP Address is needed
10223	A User name is needed
10250	Demonstration has expired
10250	Beta version has expired
10251	Invalid registration

# Error Handling

```
// An Example using FaxPack.ErrorCode  
FaxPack.AddComment("Here are some comments.")  
if FaxPack.ErrorCode=0 then  
    MsgBox("FaxPack.AddComment was successful")  
else  
    MsgBox("FaxPack.AddComment failed, error#"+Str(FaxPack.ErrorCode))  
end if
```

*For example:*

```
// An Example using FaxPack.RaiseOption=1  
FaxPack.RaiseOption=1  
try  
    FaxPack.ServerLogin("127.0.0.1", 0, "Mickey", "Disney", 1)  
    MsgBox("The Fax Server login was successful!")  
catch err  
    MsgBox("Fax Server login failed, error#", Str(err.ErrorNumber))  
end try
```

## Frequently Asked Questions (FAQ)

### **I am not specifying a cover page. Why is one being sent?**

On the FAX Server, there is a preference that will enforce "use of a cover page when none is specified". Make sure this option is turned OFF.

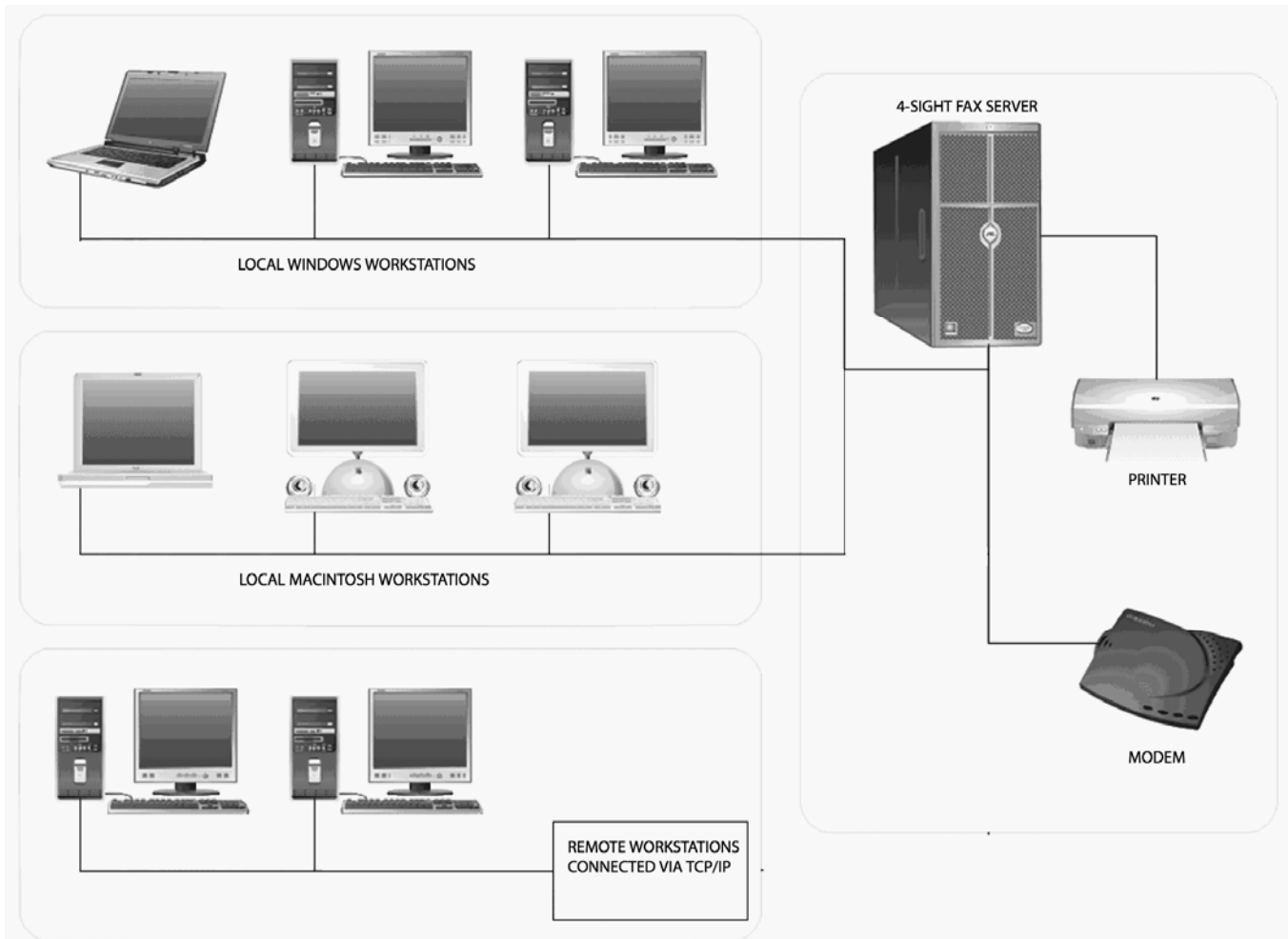
### **Why does the print job I attempt to fax, regardless of the FaxPack.switchprinter command, go to the print queue and not the fax queue?**

Be sure to always include the FaxPack.PreparePrint command (prior to issuing the REALbasic Print command), as this command is necessary for proper faxing. This will allow the FaxPack.SwitchPrinter command to work properly. Also, ensure the FaxPack.StartMessage and FaxPack.SendMessage are part of every fax job.

### **The plug-in always has to login, even on consecutive faxing attempts, and regardless of the login status of the fax client. How to I do this?**

When logging in, be sure to set the last parameter in the FaxPack.ServerLogin command to "1". This actually runs the login. If this parameter is omitted or set to "0", then the log-in will not actually occur.

# 4-Sight FAX Sample Configuration



**Figure 7:** 4-Sight FAX Sample Configuration

## Contact Information

Customer Service or Sales: 770-457-9400 or mail to: [sales@softsinc.com](mailto:sales@softsinc.com)

Technical Support: 770-457-9400 or mail to: [support@softsinc.com](mailto:support@softsinc.com)

For the latest versions of FaxPack for REALbasic, please visit: <http://www.4sightfax.com/faxpackrb.html>

For the latest versions of 4-Sight FAX, please visit: <http://www.4sightfax.com/download.html>

For 4-Sight FAX System requirements, please visit: <http://www.4sightfax.com/requirements.html>

# Index

## A

Address Book  
FaxPack.AddContact, 30  
FaxPack.DeleteContact, 31  
FaxPack.GetAddressBook, 28  
FaxPack.GetContact, 29  
FaxPack.ModifyContact, 29  
Privileges, 28

## B

Block List  
FaxPack.AddToBlockList, 32  
FaxPack.DeleteFromBlockList, 33

## C

Classes  
ContactItem, 41  
CoverPageList, 38  
ItemStatus, 40  
ServerItem, 38  
ServerList, 39  
Classes References, 38  
Command References, 34  
Constants  
convertType, 37  
DestinationKind, 36  
otherFields, 36  
Priority, 36  
Resolution, 37  
Constants References, 36  
ContactItem, 41  
Class Constants, 42  
Description, 41  
Example, 43  
Methods, 42  
Properties, 41  
Super Class, 41  
CoverPageList, 22, 38  
Description, 38  
Examples, 38  
Methods, 38  
Purpose, 38  
Super Class, 38

## E

Error  
Codes, 44  
Handling, 46

## F

FaxPack Properties, 35  
Example, 35  
Properties, 35  
Summary, 35  
FaxPack.AbandonMessage, 4, 16, 34  
Example, 16  
Purpose, 16  
Summary, 16  
Syntax, 16  
FaxPack.AddComment, 4, 9, 10, 14, 15, 34, 35, 46  
Example, 15  
Purpose, 14  
Summary, 14  
Syntax, 15  
FaxPack.AddContact, 5, 30, 31, 34, 43  
Example, 30  
Purpose, 30  
Summary, 30  
Syntax, 30  
FaxPack.AddEnclosure, 14, 34  
Example, 14  
Purpose, 14  
Summary, 14  
Syntax, 14  
FaxPack.AddRecipient, 4, 11, 12, 34, 36, 45  
Example, 12  
Purpose, 11  
Summary, 11  
Syntax, 11  
FaxPack.AddToBlockList, 5, 7, 32, 34  
Example, 32  
Login and Access, 32  
Purpose, 32  
Summary, 32  
Syntax, 32

- FaxPack.ConvertTiff, 5, 26, 34, 37
  - Example, 27
  - Purpose, 26
  - Summary, 26
  - Syntax, 27
- FaxPack.DeleteContact, 31, 34
  - Example, 31
  - Purpose, 31
  - Summary, 31
  - Syntax, 31
- FaxPack.DeleteFromBlockList, 5, 7, 33, 34
  - Example, 33
  - Purpose, 33
  - Summary, 33
  - Syntax, 33
- FaxPack.GetAddressBook, 5, 28, 34
  - Example, 28
  - Purpose, 28
  - Summary, 28
  - Syntax, 28
- FaxPack.GetContact, 5, 29, 30, 34, 41, 43
  - Example, 29
  - Purpose, 29
  - Summary, 29
  - Syntax, 29
- FaxPack.GetCoverPages, 4, 22, 34
  - Example, 22
  - Purpose, 22
  - Summary, 22
  - Syntax, 22
- FaxPack.GetFaxTiff, 5, 25, 26, 34
  - Example, 25
  - Purpose, 25
  - Summary, 25
  - Syntax, 25
- FaxPack.GetItemStatus, 4, 5, 15, 16, 34
  - Example, 17
  - Purpose, 16
  - Summary, 16
  - Syntax, 16
- FaxPack.GetQueueListing, 7, 16, 22, 34
  - Example, 24
  - Purpose, 22
  - Summary, 22
  - Syntax, 23
- FaxPack.GetTiffInfo, 5, 26, 34
  - Example, 26
  - Purpose, 26
  - Summary, 26
- Syntax, 26
- FaxPack.ModifyContact, 5, 29, 30, 34
  - Example, 30
  - Purpose, 29
  - Summary, 29
  - Syntax, 29
- FaxPack.PreparePrint, 4, 18, 19, 34, 37, 45, 47
  - Example, 19
  - Purpose, 18
  - Summary, 18
  - Syntax, 19
- FaxPack.SendMessage, 4, 11, 14, 15, 20, 21, 34, 47
  - Example, 15
  - Purpose, 15
  - Summary, 15
  - Syntax, 15
- FaxPack.ServerLogin, 3, 7, 20, 21, 22, 34, 35, 46, 47
  - Example, 22
  - Purpose, 21
  - Summary, 21
  - Syntax, 21
- FaxPack.ServerLookup, 3, 20, 21, 34, 39, 40
  - Example, 21
  - Purpose, 20
  - Return Value, 20
  - Summary, 20
  - Syntax, 20
- FaxPack.StartMessage, 4, 8, 9, 10, 11, 14, 34, 36, 45, 47
  - Cover Page Setup, 10
  - Example, 9
  - Purpose, 8
  - Summary, 8
  - Syntax, 9
- FaxPack.SwitchPrinter, 3, 4, 18, 34, 45, 47
  - Example, 18
  - Purpose, 18
  - Summary, 18
  - Syntax, 18
- FaxPack.Version, 5, 8, 34
  - Example, 8
  - Purpose, 8
  - Summary, 8
  - Syntax, 8

## I

- Image Control
  - FaxPack.ConvertTiff, 26
  - FaxPack.GetTiffInfo, 26
- ItemStatus, 40
  - Description, 40
  - Example, 41
  - Properties, 40
  - Super Class, 40

## M

- Message Control
  - FaxPack.AbandonMessage, 16
  - FaxPack.AddComment, 14
  - FaxPack.AddEnclosure, 14
  - FaxPack.AddRecipient, 11
  - FaxPack.GetItemStatus, 16
  - FaxPack.SendMessage, 15
  - FaxPack.StartMessage, 8
  - FaxPack.Version, 8

## N

- New User Setup, 7

## P

- Printer Control
  - FaxPack.PreparePrint, 18
  - FaxPack.SwitchPrinter, 18

## S

- Server Control
  - FaxPack.GetCoverPages, 22
  - FaxPack.GetFaxTiff, 25
  - FaxPack.GetQueueListing, 22
  - FaxPack.ServerLogin, 21
  - FaxPack.ServerLookup, 20
- ServerItem, 38
  - Description, 38
  - Example, 39
  - Properties, 39
  - Super Class, 39
- ServerList, 39
  - Description, 39
  - Example, 39
  - Methods, 39
  - Super Class, 39