

FaxPack™ 7.5

Plug-in for

4th Dimension & 4-Sight FAX

© 1996-2008 Soft Solutions, Inc. All Rights Reserved.

Published World-Wide by Soft Solutions, Inc.

Last updated on October 28th, 2008.

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.

4th Dimension, 4D Compiler and 4D are registered trademarks of 4D, Inc.

Soft Solutions, Inc.

2900 Chamblee Tucker Road

Building 12, Suite 200

Atlanta, GA 30341 USA

TABLE OF CONTENTS

Introduction.....	1
What's New in FaxPack Version 6, 7 and 7.5?	2
Command Overview	2
Product Registration, Server Selection and Login	3
Getting and Setting the Printer	3
Creating the Fax / Email message	4
Cover Page and Comments	4
Creating and Adding Message Attachments	4
Completing the Message.....	5
Message Details and Retrieval.....	5
Synchronizing 4D and the 4-Sight FAX Server Address Book	5
Managing Junk Faxes	6
Version Commands.....	6
Discontinued Commands	6
Command Syntax.....	6
Installation	7
What to Download	7
What to Install.....	7
Contents of the Installation	8
Platform Specific Print Issues.....	8
4D for Mac OS X.....	8
4D for Windows.....	9
4D Write 6.5 (and later) for Windows	9
4-Sight FAX Server and Configuration	10
Server Setup:.....	10
Message Control	11
SPRegister.....	11
SPGetVersion	13
SP Start Message	13
SPAddAddress.....	16
SPAddComment	19
SPEndMessage	19
SPSaveMessage	20
SPAbandonMsg	22
SPGetItemStatus	23
Message Content	26
SPAddEnclosure.....	26
SPPickEnclosure	26
SPPickMany.....	27

SPGetFilePath	28
Printer Control	30
SPSetPrinter	30
SPSwitchPrinter	30
SPPreparePrint	31
SPGetPrinter	32
SPImageControl.....	33
Server Control	35
SPSetLogin	35
SPCheckLogin	35
SPPickServer.....	36
SPGetServer.....	37
SPSetServer	38
SPGetCovers	40
SPGetServerVers	41
SPNotify	42
Queue Access	43
SPGetQueueListing	43
SPGetFaxTiff	45
SPArchiveItem	46
Image Control.....	46
SPGetTiffInfo	46
SPConvertTiff.....	47
Address Book	49
SPGetContact	49
SPModifyContact	50
SPAddContact	52
SPDeleteContact	53
SPGetAddressBook	53
Block List.....	56
SPAddToBlockList	56
SPDeleteFromBlockList	57
Command Summary.....	58
Error Codes	60
Frequently Asked Questions (FAQ).....	63
4-Sight FAX Sample Configuration.....	64
Contact Information	64
Index.....	65

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, 4D Developers can automate direct messaging capabilities for their users in their 4th Dimension application without any need for the 4-Sight FAX client.

Below is a short list of how FaxPack is being used in database applications 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

What's New in FaxPack Version 6, 7 and 7.5?

FaxPack versions 6, 7 & 7.5 introduce a number of new commands and features enabling database developers to:

- Retrieve details for entire fax / email queue listings.
- Retrieve individual fax files in any of 5 different file formats.
- Automatic convert fax Tiff files into 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 4D.
- 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.
- Version 7.5 Archive fax / email queue listings programmatically.
- Version 7.5 is compiled as a Universal Binary application.
- Version 7.5 is 4D v11 (Mac/Windows) and Leopard compatible (for Macintosh).

* Prior to 4-Sight FAX and FaxPack version 7/7.5, the 4-Sight FAX Server software utilized a number identification system that was dynamic. Under prior versions, fax identification numbers changed each time the 4-Sight FAX server was restarted. Under 4-Sight FAX 7 and FaxPack version 7, a permanent unique numbering system is in place. This permits users to now confidently create, retrieve and delete faxes while providing database users permanent, accurate queue information.

Command Overview

Before describing the FaxPack commands in detail, it will be helpful to overview the purpose, use and basic steps to follow in programming your fax solution. The 4-Sight FAX server permits a 4-Sight FAX v7 client and a 4th Dimension client using FaxPack v7 to run concurrently on the same machine with the same user[name] using the same TCP/IP address.

Product Registration, Server Selection and Login

The **SPRegister** command must first be called to allow uninterrupted access to the FaxPack development commands (without time limits). The 4-Sight FAX server that transmits the completed message must be identified in your session before a message can be sent.

The FAX server can be identified explicitly using the **SPSetServer** command. Alternatively, the **SPPickServer** command can be used to allow selection of the FAX server. The **SPSetLogin** command establishes the User and password information and is a required command.

The **SPSetLogin** command does not validate the information nor actually log-in the user. The **SPSetLogin** parameter values, however, are made available to over a half dozen FaxPack commands that require log-in access dynamically. Each of these commands logs in and logs out of the server using the **SPSetLogin** parameter values. A valid user name and password are checked against the 4-Sight FAX Server before any faxes are created and are a log-in requirement with version 7. The **SPCheckLogin** command is optional, and validates the User and password information and ensures the log-in information is valid.



IMPORTANT: With the 4-Sight FAX Client software, users login and remain logged in until they log out. With FaxPack, however, users are logged in as certain FaxPack commands are executed and logged out as FaxPack commands are completed. The same User from the same IP address can be simultaneously logged in more than once through the 4D Client and FaxPack.

Getting and Setting the Printer

The **SPGetPrinter** and the **SPSetPrinter** commands are to be used in conjunction with each other. The **SPGetPrinter** command obtains the current printer string of the printer to be used in the **SPSetPrinter** command (preferably of 4-Sight FAX). The **SPSetPrinter** command allows the User to specify the 4-Sight FAX v7 print/ fax driver. It does not actually change the current printer, it simply tells FaxPack to use this Fax Printer driver when it is requesting to fax.



IMPORTANT: Users are advised to first manually set or establish the current print driver to be the 4-Sight Fax Print driver before issuing the **SPGetPrinter** command. This will allow the **SPGetPrinter** command to work as it was designed, obtaining the necessary 4-Sight Fax Print driver string. The **SPGetServer** command can be used to obtain fax server details for use with **SPSetServer** command in subsequent sessions.

The **SPSwitchPrinter** command allows you to procedurally alternate between the 4-Sight FAX printer driver and the user's selected printer, without opening the Mac OS 9.2.x "Chooser", the Macintosh OS X "Print Center" or the Windows printer Control Panel. When running under Mac OS X, version 10.2.2 or later must be installed in order to use the "**SPSwitchPrinter**" command. The **SPPreparePrint** command communicates directly with the 4-Sight FAX v7 print driver. This command tells FaxPack to capture the subsequent 4D print job into a temporary Tiff file for transmission to the 4-Sight FAX server.



IMPORTANT: Since 4D 2003, 4D provides the “Set Current Printer” Command, which functions similarly to the **SPSetPrinter** command. In addition, the “Printers List” 4D command is available which returns an array of all available printers. 4D Write (and 4D) works well with these commands and they can be used effectively in your 4D application.

Creating the Fax / Email message

A fax / email message (message) is built with several FaxPack calls. To begin, the 4D developer should call the **SPStartMessage** command to provide basic sender and sending information. A “message” in this context refers to either a fax or email, generated through a 4D 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 **SPAddAddress** 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.

Cover Page and Comments

The **SPGetCovers** command will provide the 4D Developer with the names of the Fax Server cover Pages (up to 14). The **SPAddComment** command can optionally be used to specify a text comment for the fax cover page of the current fax message. This command also allows you to specify the resolution to be used when imaging the cover page.

Creating and Adding Message Attachments

The **SPSaveMessage** command allows the user to save a permanent TIFF file for subsequent use as an attachment. The “**SPSaveMessage**” command creates a Tiff file (a “.4TF” file) used by 4-Sight FAX v7. The **SPSaveMessage** requires only a single issuance of the **SPPreparePrint** command. This command allows the user to create and name a Tiff file programmatically for subsequent attachment within a FaxPack initiated fax message. The order of events for creating a file from FaxPack is now as follows: An **SPPreparePrint** command, followed by A) the 4D print job (Print form, etc.) and B) lastly, the **SPSaveMessage** command. After creating a “.4TF” file, this file can be attached to any 4-Sight FAX v7 message using the **SPAddEnclosure** command.

4D Developers can provide file selection through the customizable **SPGetFilePath** command. This command establishes a default file path prompting Users from an appropriate folder. This is valuable as fax enclosures can be optionally generated from within 4D using the **SPSaveMessage** command (discussed below). Other Macintosh application documents may

be imaged and printed to disk using the 4-Sight FAX Print Driver or via the **SPSaveMessage** command. These TIFF format files may be enclosed explicitly by using the **SPAddEnclosure** command.

You can ALSO allow the 4D user to select valid enclosure files using the **SPPickEnclosure** command. Alternatively, with the **SPPickMany** command, 4D Developers can provide Users with a multi TIFF-file picker for multiple file selection from a single file dialog. Both commands, **SPPickEnclosure** and **SPPickMany**, restrict file display to show only the native TIFF format “.4TF” file types for v7.



IMPORTANT: The **SPAddEnclosure** does not restrict the developer from specifying an invalid file type, so appropriate care must be taken.

Completing the Message

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



IMPORTANT : Every call to **SPStartMessage** must be balanced with a final call to either **SPEndMessage**, **SPSaveMessage** or **SPAbandonMsg**.

Message Details and Retrieval

The **SPGetItemStatus** command allows the user to retrieve the fax status, date and time sent, etc. for any fax. The **SPGetFaxTiff** allows the 4D application to retrieve any fax file and is most useful for databases providing viewing capabilities. The **SPGetTiffInfo** command provides the developer with the number of pages for the fax just retrieved. The **SPConvertTiff** command converts one or more pages from a Fax Tiff file into any of 4 different file formats, including PDF, Pict, PNG, BMP and Tiff. When using the **SPConvertTiff** command, files can be converted and saved to the applications local hard driver.

Synchronizing 4D and the 4-Sight FAX Server Address Book

New to FaxPack version 7, the **SPGetContact** command provides the 4D Developer a tool for retrieving 22 fields of information for any Contact within the FAX Server address. The

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 **SPAddContact** 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 **SPModifyContact** command. Lastly, the **SPGetAddressBook** 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 4D database.



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

Managing Junk Faxes

Also new to FaxPack version 7 are commands for adding and removing fax numbers and companies from the Junk Fax List, maintained by the 4-Sight FAX Server. The new commands are **SPAddToBlockList** and **SPDeleteFromBlockList**.

Version Commands

The **SPGetVersion** command, new to version 7, returns the string of the current FaxPack version number. The **SPGetServerVers** command identifies the current version of the 4-Sight FAX Server.

Discontinued Commands

The **SPNotify** command is no longer functional under 4-Sight FAX Server version 7, although it remains part of the available command set for legacy purposes.

Command Syntax

Many of the FaxPack commands use string and/or text parameters. Where a specific string length is not mentioned, the 4D Developer can use a string size up to 255 characters or a text field.

Installation

What to Download

Download either:

1. FaxPack for 4D running on Macintosh (4D or 4D Server).
2. FaxPack for 4D running on Windows (4D or 4D Server).

Based on the version of 4D you are operating, please pick the appropriate download file based on the target platform.

- 4D Version 6.0 -> 6.5
- 4D Version s 6.7, 6.8,
- 4D 2003
- 4D 2004 & 4D v11.

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/v7.5 print driver for Macintosh OS X and the Windows print driver. Under Macintosh OS 9.2.2, however, the user can simply copy the print driver provided into the extensions folder located within the System Folder and then restart the machine. This procedure ensures the Macintosh system will recognize this 4-Sight FAX print driver.

For 4D 2004 & 4D v11, installation is achieved by copying the "FaxPack.bundle" file to your 4D application's "Plugins" folder. The "FaxPack.bundle" file contains both the Macintosh & Windows plug-ins.

Installation of the FaxPack plug-in is achieved by copying the plug-in file to your 4D application's Mac4DX or Win4DX folder based on the installed operating system:

Under Mac OS X, install the "FaxPack.4CX" file into your application's Mac4DX folder.

Under Mac OS 9.2.2, install the "FaxPack[.4DX]" file into your application's Mac4DX folder.

Under Windows, install the "FaxPack.4DX" and the "FaxPack.rsr" file into your application's Win4DX folder.

After the FaxPack plug-in is installed, a new set of commands for working with the 4-Sight FAX Server will appear in the standard 4D method editor window. Review the 4th DIMENSION

documentation for special instructions on: a) how plug-ins are handled when you intend to compile your program, b) how to install it for use by multiple users via 4D Server, or c) how to use an Intel-based PC as your 4D application server.

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 (4D or 4D Server). There is also one download for Windows (4D or 4D Server).

The following items are contained in the FaxPack 7.5 Folder:

The FaxPack Demo v7.5 Folder contains:

- FaxPack demo database files
 - FaxPack v7.5 Release Notes & FaxPack Order form
 - Mac4DX and Win4DX folders (for 4D 6.0 to 4D 2004)
 - Plugins folder (for 4D 2004 & 4D v11)
 - FaxPack v7.5 Manual
 - Installer for the 4-Sight FAX (v7.0 or later) print driver (for installation on client machines and machines where FaxPack applications operate).
- The FaxPack Demo folder contains freely distributable source code.

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

Platform Specific Print Issues

4D for Mac OS X

Under Macintosh OS X, 4D establishes a default printer at application start-up time and may disregard programmatic changes to this default printer without one-time user intervention. To circumvent this issue, the 4D "Print Settings" command must be issued to establish 4-Sight FAX as the new default print driver after launching 4D. For this reason, it will be necessary for 4D developers to present the 4D "Print Settings" dialog to allow the User to select the 4-Sight FAX printer driver before attempting to fax. It is recommended the call to 4D Print Settings be made prior to calling the FaxPack **SPStartMessage** command.

Under OS X, the print settings command first brings up a page setup dialog followed by a print settings dialog. In the "Print settings" dialog, the User must select the 4-Sight FAX print driver in order for 4D to respect the change in printer drivers.

The FaxPack command **SPSwitchPrinter** correctly switches the Print Driver to 4-Sight FAX from the OS X Print Center, however, depending on the versions ranging from 6.0 through 2004, 4D may not honor this change. Therefore, it may be necessary for the user to change to the 4-Sight FAX print driver manually before attempting a fax job.



IMPORTANT: 4D's "Set Current Printer" Command and the 4D "Printers List" command are available in 4D 2003 and 2004. 4D Write and 4D work well with these commands and can be used effectively in your 4D application.

4D for Windows

When using the 4D Print Form command, 4D fails to properly toggle to the new printer until after a first Print job has been executed. For this reason, the 4D FaxPack Demo for Windows calls a 4D method to generate a dummy print job. This method permits the subsequent "real" fax job to process correctly. Please include a comparable method in your 4D fax application under Windows to ensure this desired behavior.

4D Write 6.5 (and later) for Windows

4D Write acts like a stand-alone application. For this reason, 4D Developers should follow the programming technique provided within the FaxPack Demo for 4D for Windows when attempting to fax from a 4D Write area. In this Fax Demo, the Letters output form method runs 4D code to switch the default printer to the 4-Sight FAX printer before loading the 4D Write area. When the 4D Write area is subsequently loaded in the Letters input form, this process signals to 4D that 4-Sight FAX is the default printer. Soft Solutions has found this methodology to be reliable when faxing from 4D Write under 4D for Windows, version 6.5, 6.8, 2003 and 2004.

4-Sight FAX Server and Configuration

Server Setup:



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

Printing:

- Use General Settings
- No Printing
- Direct Printing

Phone Card:

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

Command Descriptions

Message Control

SPRegister

Syntax: *SPRegister(DeveloperName:S; Mac RegNumber:S;Win RegNumber:S;Deployment Lic#:S): ResultCode:S*

Purpose:

The SPRegister command in FaxPack version 7 provides the 4D Developer a tool to register the FaxPack plug-in for unrestricted usage. This command must be called with at least two string parameters; the licensed Developer name and the Registration Number for Macintosh and/or Windows. Each string parameter is separated by a semicolon and should be enclosed within quotes. The SPRegister command should be called prior to issuing any FaxPack commands. When the command is called and passed valid parameters, it will return a string that identifies the version of the FaxPack package. This string will have the same value as displayed by the “versions: field” of the Finder’s Get Info box for the package plug-in file (e.g., “7.0”). An invalid, empty or missing registration from the requesting machine’s platform will return the string “INVALID REGISTRATION” or “INVALID DEPLOYMENT” and disable the plug-in area after 45 minutes. When this happens, the first call will cause a “demo” dialog to appear stating the program is for evaluation only. Under “demonstration” mode, the FaxPack commands will function normally for 45 minutes and cease to function afterwards.



IMPORTANT: The “INVALID DEPLOYMENT” message appears along with the serial number of the 4D Server where the current 4D application is running. This 4D Server serial number is required in order to obtain an individual 4D server deployment license number from Soft Solutions. Alternatively, the 4D Developer can obtain this serial number by issuing the 4D command “Get Serial Information” from 4D Client. With version 7, it is recommended the User compare the string returned with the SPGetVersion command and the SPRegister command “If(SPRegister=SPGetVersion) then...”

An OEM deployment license is based on the Licensee name and does not require an individual 4D Server serial number. A valid registration returns the current FaxPack version number (as found in the “Get info” window) of FaxPack.

Parameters:

Name	Explanation
DeveloperName	A String value. This value can only be provided by Soft Solutions, Inc. or an authorized FaxPack reseller.
Mac RegNumber	A String value. This value works in conjunction with the Developer Name parameter passed in parameter 1. For a Windows only setup,

	Developers can code this as a null string. This code can only be provided by Soft Solutions, Inc. or an authorized FaxPack reseller.
Win RegNumber	A String value. This value works in conjunction with the Developer Name parameter passed in parameter 1. For a Macintosh only setup, Developers can code this as a null string. This code can only be provided by Soft Solutions, Inc. or an authorized FaxPack reseller.
Deployment Lic#	A string expression providing the 4D Client Server deployment license code. For non-OEM licensing (based on an individual 4D Server deployment), Soft Solutions requires the 4D Developer to initially provide them with the 4D Server number where FaxPack will be deployed. This 4D server number is numeric, between 6 and 10 numbers long and is the basis for generating the deployment license code. For OEM deployment (aka vertical market application or “blanket” deployment), the deployment license is created from the Developer Name in lieu of the 4D Server Serial number.

Return Value:

Name	Explanation
ResultCode	A string value. If the operation is successful, the current version of FaxPack is returned. If invalid then either “Invalid Registration” or “Invalid Deployment” will be returned.

Example:

\$Version:=SPRegister(“Name”;“MacCode”;“WinCode”;“DeploymentCode”) ‘Issued by SSI

SPGetVersion

Syntax: *SPGetVersion: ResultCode:S*

Purpose:

The SPGetVersion command, new to version 7, returns the string of the current FaxPack version number. Users can check the value returned from this command with the value returned from the SPRegister command to simplify product registration coding.

No Parameters Return Value:

Name	Explanation
ResultCode	A string value. If the operation is successful, the current version of FaxPack is returned. If an error occurs, a non-zero error code value is returned.

Example:

\$Version:=SPGetVersion

SP Start Message

Syntax: *SPStartMessage(From:S; Subject:S; CoverPage:L; Reserved:I): ResultCode:I*



Purpose:

The SPStartMessage command begins the process of building a new fax 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 banner.

You always begin the process of creating a fax message with a call to SPStartMessage, and then use other FaxPack commands to define the content of the message. The message is then either sent to the server with a call to the SPEndMessage command, sent to a file with SPSaveMessage or cancelled with a call to SPAbandonMsg. Please note; the "From" parameter must contain a valid User name stored on the 4-Sight FAX Server, otherwise the Fax message will be rejected.

After you call SPStartMessage, a corresponding SPEndMessage, SPAbandonMessage or SPSaveMessage must be issued prior to any additional SPStartMessage commands being issued. Any of these three commands will discard any outstanding message information. Developers should issue the SPAbandonMessage whenever an uncorrectable error occurs.

Parameters:

Name	Explanation
From	<p>A string expression that will appear in the From field on the fax header page.</p> <p> IMPORTANT: In version 7.0 of 4-Sight FAX, it is required that this parameter contain a valid User name as defined in the 4-Sight FAX Server list of User names.</p>
Subject	<p>A string expression containing a brief description of the fax content that will appear in the Subject field on the fax header page</p>
CoverPage	<p>A long integer expression that specifies the fax cover page profile banner line format to be used for this message. A value of 1 will select the first header page, 2 will select the second, and so on. The value 15 (or 0) indicates that no header page is required.</p> <p> IMPORTANT: If the SPAddComment command is used, yet no cover page is specified here (a 0), the default cover page will be used when this option (<i>use default cover page when none selected</i>) is established as the default within the 4-Sight FAX Server preferences.</p>
Reserved	<p>An integer expression that is not used by this release. This parameter is reserved for future use; pass a value of zero for now.</p>

Return Value:

Name	Explanation
ResultCode	<p>If the memory allocation for the new message is successful, zero is returned. If an error occurs, a non-zero error code value is returned.</p>

Example:

```
$ResultCode:=SPStartMessage("From";"Subject";CoverPage)
```

Server 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 SPStartMessage, yet issues the SPAddComment with comments, a cover page will be used. The cover page chosen will be determined by the settings on the dialog below. If the checkbox within 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

Cancel

OK

SPAddAddress

Syntax: *SPAddAddress(Name:S; Company:S; Fax Number:S; IsUrgent:I; DestKind:I; SendDate:D; Send-Time:H; Country Code:S; Area Code:S; Other Fields:X): ResultCode:I*

Purpose:

The SPAddAddress command is used to add a destination address to a message. A message must have at least one destination address, but there is no limit to the number of additional destination addresses a message may have.



IMPORTANT: You must call the SPAddAddress command at least once, after calling SPStartMessage and before calling SPEndMessage.

Parameters:

Name	Explanation
Name	A string expression providing the name of the recipient of this transmission. This information is not used in the actual computer addressing.
Company	A string expression providing the Company (or postal address) of the recipient, used on the cover page of a fax.
Fax Number or Email Address	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.
IsUrgent	An integer expression providing the desired priority for the transmission. Zero (0) indicates a Normal Priority, and a value of 1 indicates Urgent Priority
DestKind	This specifies 1 of three alternatives: 1 = Fax, 2 = Fax with Smart Dial and 3 = Email.
SendDate	A date expression specifying the desired message send date. If this value is zero or less than current date, the server will process the file at the first possible opportunity. If it resolves to some time in the future, transmission will be delayed until that time.

SendTime	A time expression specifying the desired message send time. If this value (in conjunction with the SendDate:D parameter above) resolves to some time in the past, the server will process the file at the first possible opportunity. If it resolves to some time in the future, transmission will be delayed until that time.
CountryCode	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.	A string expression used if "DestKind" is 2 for SMART Dialing. The value is the dialing string required for the recipient's area code.
Otherfields	(Optional). A text array containing 17 elements populated from the 4-Sight FAX Address Book. {1} Salutation, {2} First Name, {3} Last Name, {4} Title, {5} Category, {6} Alternate, {7} Voice, {8} Email, {9} Address1, {10} Address2, {11} City, {12} State/Province, {13} Country, {14} Postal Code, {15} Custom 1, {16} Custom 2 & {17} Custom 3

Return Value:

Name	Explanation
ResultCode	If the operation is successful, a value of zero is returned. If an error occurs, a non-zero error code value is returned. If the Fax # or Email Address is empty, an Error code of 10217 is returned.

Example:

```
$ Result:= SPAddAddress("Recipient Fullname";"Company";"Fax Number";0; 1; $SendDate; $SendTime; "Country Code"; "Area Code";otherFields)
```

Server Setup:

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: United States

Country code: 1

Area code: 212 Area Code Rules...

International dial code: 011

Long distance dial code: 1

Smart dial
 Always dial long distance
 Always dial International

Cancel OK

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)

<p>Outside my area code: Do not dial 1 for the following area codes:</p> <div style="border: 1px solid gray; height: 100px; width: 100%;"></div> <div style="display: flex; justify-content: flex-end; gap: 5px;">AddDelete</div>	<p>Within my area code: Dial 1 for the following prefixes:</p> <div style="border: 1px solid gray; height: 100px; width: 100%;"></div> <div style="display: flex; justify-content: flex-end; gap: 5px;">AddDelete</div>
---	---

Cancel OK

SPAddComment

Syntax: *SPAddComment(Comment:T; Resolution:L): ResultCode:I*

Purpose:

The SPAddComment command can be used to specify a text comment and the fax imaging resolution for a message's cover page. You may call this command only after calling SPStartMessage and before calling SPEndMessage. You may only specify one cover-page comment per message; additional text may be included with the message as a text file enclosure.



IMPORTANT: Issuing the SPAddComment 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 SPStartMessage command. 4-Sight FAX, beginning in version 4, stores and utilizes a default cover page stored in a Macintosh resource.

Parameters:

Name	Explanation
Comment	A text expression providing the comment text to be displayed on the header page of a fax or as part of the body of a telex.
Resolution	A long integer expression that specifies the resolution required for the fax header page: 1 = standard resolution, 2 = fine resolution.

Return Value:

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

Example:

\$ Result:= SPAddComment(Comment;Resolution)

SPEndMessage

Syntax: *SPEndMessage(Message ID:S; SingleUser:I; AllowPicker:I): ResultCode:I*

Purpose:

The SPEndMessage 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. The Message ID parameter is assigned to a specific outgoing fax by the 4-Sight FAX v7 FAX server. After the fax message is generated and returned within the "Message ID" value, it can be later referenced using the SPGetItemStatus call to obtain the current status of that fax.

Parameters:

Name	Explanation
Message ID	A string variable that receives the Message ID of the current fax message after it arrives at the fax server and is assigned this unique number. The "Message ID" is unique and the integrity of this Message ID's uniqueness is maintained at the version 7 FAX server.
SingleUser	An integer expression which is no longer user for legacy purposes, specify 0.
AllowPicker	This parameter is ignored in version 7. Developers should set this value to 0. It is unlikely this request would be issued unless the Server had not been chosen prior to this command being issued. Please ensure the Server is set beforehand.

Return Value:

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

Example:

```
$ ResultCode:=SPEndMessage ($messageID;0;1)
```

SPSaveMessage

Syntax: *SPSaveMessage(FilePath:S; SFPrompt:S; SFDefault:S): ResultCode:I*

Purpose:

The SPSaveMessage command enables you to create a fax "message" file(s) for use in later fax messages as an enclosure file. Similar to SPEndMessage, the SPSaveMessage command

is used to finalize the current message. Consequently, you should use the `SPSaveMessage` command instead of `SPEndMessage`, to signify you wish to dispense with the current message by saving it to a “message” file instead of sending it to the Fax Server for transmission.



IMPORTANT: DO NOT use both commands, `SPSaveMessage` and `SPEndMessage`, for the same fax as they are mutually exclusively. Because this message is a file and there are no recipients (i.e., destination addresses), there is no need to specify any destination address (`SPAddAddress`) or cover page information (`SPAddComment`).

Utilizing the `SPPickMany` and `SPPickEnclosure` commands, a User can enclose these “message” files in a subsequent fax transmission.

Parameters:

Name	Explanation
FilePath	A string variable specifying the full path to the message file you wish to create. If you wish to allow the user to specify the location of the new file, assign a null string to this variable before calling the command; in this case, the command will display a standard file package “Save As” dialog to allow the user to enter a file name and select the folder in which to save the file. When the command has completed, the full path to the file chosen by the user will be assigned to this variable.
SFPrompt	A string expression that specifies a display prompt for the user on the standard file package dialog. If a path value is assigned to FILEPATH before the command is called, this parameter is ignored.
SFDefault:S	A string expression that specifies a default folder and/or filename for the user when the standard file package dialog is displayed. If a path value is assigned to FILEPATH before the command is called, this parameter is ignored. If you pass the full path to an existing folder here, the standard file dialog will pre-select that folder for the user and display its contents in the catalog list area, thereby “suggesting” that folder as the location for storing the new file in its filename edit field. If you pass the full path <u>to a file</u> in an existing folder, the dialog will both pre-select the folder and enter the filename as the default name.

Return Value:

Name	Explanation
------	-------------

ResultCode	If the operation is successful, a value of zero is returned. If an error occurs, a non-zero error code value is returned. If the User cancels the standard file dialog, a value of -128 is returned.
------------	--

Example:

`$Err:=SPSaveMessage ($fullFilePath;$SFprompt;$SFDefaultFilePath)`

SPAbandonMsg

Syntax: *SPAbandonMsg*

Purpose:

This command disposes of all memory allocated when creating a message. It is typically called if an error condition was returned by one of the other functions, or if the application has to abandon the message for any reason. No Parameters exist for this command and therefore, no Result codes are returned.

Parameters:

Name	Explanation
N/A	No Parameters exist for this command

Return Value:

Name	Explanation
ResultCode	This is not a function call, therefore no error code values are returned.

Example:

SPAbandonMsg `No parameters are required

SPGetItemStatus

Syntax: *SPGetItemStatus(UniqueMessage ID:S; QueueTypeOrIndex:I; PageCount:I; Status:S; From:S; To:S; Subject:S; Address:S; Number:S; NextSendTime:H; IsDone:I; IsPrinted:I; IsRejected:I;IsDeleted:I): ResultCode:I*

Purpose:


The SPGetItemStatus command allows users to determine 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 SPGetQueueListing command which allows developers to obtain the Unique ID for any fax.



IMPORTANT: The QueueTypeOrIndex (2nd) parameter should be evaluated initially to determine the nature of the value within the UniqueMessageID parameter.

Parameters:

Name	Explanation
UniqueMessage ID	When the QueueTypeOrIndex parameter value is greater than 0, this parameter contains a string expression that provides the unique Message ID of the completed outgoing fax message file, as returned by the Message ID parameter of the SPEndMessage call. When the QueueTypeOrIndex parameter value is -1, this value reflects the unique fax ID assigned to the outgoing fax item by the 4-Sight FAX server. When the Queue Type or Index value is -2, it reflects the unique fax ID assigned to the incoming fax item by the 4-Sight Fax server.
QueueTypeOrIndex	An integer expression identifying either: A) which queue the fax comes from, or B) the fax index for a fax broadcast. Specify -1 for the outgoing queue, -2 for the incoming queue or the index number for a the specific recipient for which you wish to obtain status information. Recipients are identified by the sequential order in which they are added to the message; for example, to obtain the status of the transmission to the 3 rd recipient of the specified message, pass a 3 in this

	parameter. This parameter is typically used in a facsimile or email "Broadcast" where a static message is sent to one or more recipients. Specify 1 as default for individual fax messages.
PageCount	An integer variable; when the call is completed, this variable will be assigned the value of the number of pages in the specified transmission.
Status	A string variable; when the call is completed, this variable will be assigned a value specifying the current status of the specified transmission. Responses include 1). "Sent" 2). "Max Retries" 3). Failed 4). "Prep n/n" where the numerator is the page being prepared and the denominator is the total number of pages to be prepared. 5). "Image n/n" and 6) "Failed".
From	A string variable; when the call is completed, this variable will be assigned the value of the From field on the fax cover page header of the specified transmission.
To	A string variable; when the call is completed, this variable will be assigned the value of the recipient name for the specified transmission.
Subject	A string variable; when the call is completed, this variable will be assigned the value of the Subject field on the fax cover page of the specified transmission.
Address	A string variable; when the call is completed, this variable will be assigned the value of the recipient address for the specified transmission.
NumberOrEmail	A string variable; when the call is completed, this variable will be assigned the value of the recipient fax number or email address(es) for the specified transmission.  IMPORTANT: Because the developer may pass multiple email addresses in the SPAddAddress command, each address shown in

	this parameter (for an outgoing email only) would be separated by a comma.
NextSendTime	A time variable that is not functional in FaxPack v7, although it is still in place to simplify conversion of legacy FaxPack applications.
IsDone	An integer variable; when the call is completed, this variable will be assigned a value of 1 if the specified transmission has been processed, and 0 if not.
IsPrinted	An integer variable; when the call is completed, this variable will be assigned a value of 1 if the specified transmission has been printed, and 0 if not.
IsRejected	An integer variable; when the call is completed, this variable will be assigned a value of 1 if the specified transmission has been rejected, and 0 if not
IsDeleted	An integer variable; when the call is completed, this variable will be assigned a value of 1 if the specified transmission has been deleted, and 0 if not.

Return Value

Name	Explanation
Result Code	If the operation is successful, a value of zero is returned. If an error occurs, a non-zero error code value is returned. Possible non-zero responses include “-1”: Demo Expired, “-43”: File Not Found, and “-128”: User Cancelled Operation.

Example

```
$err:=SPGetItemStatus([FaxStatus]ID_FaxMessage,[FaxStatus]QueueTypeOrIndex,$PageCount,$Status,$From,$To,$Subject,$Address,$FaxNumOrEmail,$NextTime,$IsDone,$IsPrinted,$IsRejected,$IsDeleted)
```

Message Content

SPAddEnclosure

Syntax: *SPAddEnclosure(Filename:S): ResultCode:I*

Purpose:

The SPAddEnclosure command allows you to attach a specified file to the current message as an enclosure. You may call this command only after calling SPStartMessage and before calling SPEndMessage. This command relies on the attached files being the supported file type for 4-Sight FAX v7, with a file type of “.4TF”.

Parameters:

Name	Explanation
Filename	A string expression providing the full path to the enclosure file to add.

Return Value:

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

Example:

For (\$i;1;Size of array(aEnclFile)) `Arrays built with documents full path name.

\$Err:=SPAddEnclosure (aEnclFile{\$i})

End for

SPPickEnclosure

Syntax: *SPPickEnclosure(Filename:S): ResultCode:I*

Purpose:

The SPPickEnclosure is used to display a standard “Get File” dialog, allowing the user to select a suitable file for enclosing (select “.4tf” files for 4-SIGHT FAX TIFF). When the user selects a file, it is automatically attached the current message. SPPickEnclosure can be called at anytime and is independent of the actual fax messaging and transmission.

Parameters:

Name	Explanation
Filename	A string variable which upon return will be assigned the full path of the file selected by the user.

Return Value:

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

Example:

*\$Err:=**SPPickEnclosure**(\$Filename)`Where \$Filename contains the full path name of the selected file.*

SPPickMany

Syntax: *SPPickMany(DefaultPath:S;FilesArray:X): ResultCode:I*

Purpose:

The SPPickMany command is an enhanced version of the SPPickEnclosure command. It provides the end user with a customized standard file dialog allowing them to navigate their volumes and folders and select one or more enclosure files for subsequent transmission. Similar to the SPPickEnclosure command, SPPickMany can be called at anytime and is independent of the actual fax messaging and transmission.

Parameters:

Name	Explanation
DefaultPath	A string expression that specifies a folder that is to be initially displayed by the standard file dialog.
FilesArray	A text array (pre-declared with 4D's ARRAY TEXT command) that lets your application know which enclosure files were selected by the user. When the command is completed, this array will have one element for each file selected, and that element will contain the value of

	the full path to the file. If the user cancels the dialog, the array will contain 0 elements. The Files Array is cleared each time the SPPickMany command is called.
--	--

Return Value:

Name	Explanation
ResultCode	If the operation is successful, a value of zero is returned. If an error occurs, a non-zero error code value is returned. If the User cancels the standard file dialog, a value of -128 is returned.

Example:

`Where FilesArray is an array containing the full path names of all the files selected.

`$Err:=SPPickMany($DefaultPath;FilesArray)`

SPGetFilePath

Syntax: `SPGetFilePath(Filename:S): ResultCode:I`

Purpose:

The SPGetFilePath command invokes a Standard Get File dialog, allowing the user to select a file to use as an enclosure. Prior to calling the command, the Developer may populate the Filename parameter with a valid full path name so the standard file dialog will initially display the contents of the specified folder. If a null string is passed, the dialog will display the current default directory. Assuming a file is then selected, the Filename parameter will return the full path to this file. You may store this path and pass it to SPAddEnclosure at some later point in time. Since SPGetFilePath does not automatically add the selected file to the current message as an enclosure, (i.e., it does not involve itself in the message build/send process), it may be called at any time.



IMPORTANT: The SPGetFilePath command does not filter for valid file attachment types. When using this command, please ensure the User is not in a situation where they may inadvertently choose an invalid file type and, subsequently, be in position to attach this invalid file type to an outgoing fax message.

Parameters:

Name	Explanation
Filename	A string variable which upon return will be assigned the full path of the file selected by the user.

Return Value:

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

Example:

`DefaultFilePath contains full path presented to user within a standard Get File dialog.

*\$Err:= **SPGetFilePath**(\$DefaultFilePath)*

Printer Control

SPSetPrinter

Syntax: *SPSetPrinter(DeviceName:S): ResultCode:I*

Purpose:

The SPSetPrinter command enables you to specify the name of the Fax-Server's "alternate printer" device file. The alternate printer mechanism enables you to procedurally activate and deactivate the production of fax enclosure files during the printing operations performed by your 4D application, without requiring your user to manually switch the selected printer via the OS printer tools.

By default, the alternate printer device file name is "4-Sight FAX". If the 4-Sight FAX driver file has been renamed, you can use this command to identify the new file name. You may then use the SPSwitchPrinter command to alternately switch the user's currently active printer device between that (selected by the OS) and the specified alternate printer device.

Parameters:

Name	Explanation
DeviceName	A string expression providing the name of the alternate fax printer driver file.

Return Value:

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

Example:

```
$Err:= SPSetPrinter($DeviceName)
```

SPSwitchPrinter

Syntax: *SPSwitchPrinter(Mode:I): ResultCode:I*

Purpose:

The `SPSwitchPrinter` command toggles or sets the printer device to be used for subsequent print jobs between the default printer and the Fax Server's alternate printer device. The alternate printer device's name is "4-Sight FAX" by default, but can be reassigned using the `SPSetPrinter` command.

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.

Parameters:

Name	Explanation
Mode	An integer expression that specifies the desired operation Return Value: 0 = toggle printers, 1 = select the user's normal printer, 2 = select the alternate (fax) printer.

Return Value:

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

Example:

```
C_Integer(RegularPrinter:FaxPrinter)
$RegularPrinter:=1
$FaxPrinter:=2
$Err:= SPSwitchPrinter($FaxPrinter)
```

SPPPreparePrint

Syntax: `SPPPreparePrint(Resolution:L): ResultCode:I`

Purpose:

The `SPPPreparePrint` command sets the resolution. In addition, **it is required** for every 4D print job and allows you to automate the generation of fax image files for controlled transmission. A *FaxPack 7.5*

flag is set in the control structure and the current date and time is stored. This information is used by the 4-Sight FAX Printer Driver on the next fax print. The flag suppresses the user interface and the image is printed directly to disk, adding it automatically as an enclosure to the current message. It also accepts the required fax resolution for this enclosure as a parameter.

Parameters:

Name	Explanation
Resolution	A long integer expression that specifies the resolution required for the fax image file: Return Value: 1 = standard resolution, 2 = fine resolution.

Return Value:

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

Example:

`$Err:= SPPreparePrint(2) `Specify fine resolution`

SPGetPrinter

Syntax: *SPGetPrinter(DeviceName:S): ResultCode:I*

Purpose:

The SPGetPrinter command enables you to retrieve the name of the current default printer. This command is useful for determining the specific printer name of the 4-Sight FAX printer driver. Before issuing this command to obtain the printer driver name for 4-Sight FAX, it will be necessary to first establish the 4-Sight FAX printer driver as the current default printer.



IMPORTANT: On the Mac OS, the printer device file name for 4-Sight FAX is “4-Sight FAX.” If the 4-Sight FAX driver file has been renamed or altered by the OS, you can use this command to identify the new file name.

Parameters:

Name	Explanation
DeviceName	A string expression providing the name of the alternate fax printer driver file.

Return Value:

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

Example:

`$Err:= SPGetPrinter($Devicename) `Specify 4-Sight FAX Printer Driver`

SPIImageControl

Syntax: *SPImageControl(Defaults:l;Scaling Percent:l;Starting Page Number:l;Ending Page Num-ber;l;Grayscale;l;PaperType;l;Preview;l): ResultCode:L*

Purpose:

The SPIImageControl command allows the 4D Developer to control the print settings through the 4-Sight FAX Print Driver. It is necessary to first call the SPPpreparePrint command before calling the SPIImageControl command.

Parameters:

Name	Explanation
Defaults	An integer that resets all settings to their defaults, ignoring the remaining 6 parameters when it is set to 1. Set to -1 to allow use of the other 6 parameters.
Scaling Percent	An integer value that reflects the percent of page scaling to occur. Valid percents range from 25 to 500).
Starting Page #	An integer value that reflects the starting page number to be generated into a Fax file. A value of 0 in this field and in Ending

	Page# will print all document pages.
Ending Page #	An integer value that reflects the ending page number to be generated into a Fax file. A value of 0 in this field and in Starting Page# will print all document pages.
Grayscale	An integer value to indicate whether grayscale is used; 0 = no, 1 = yes.
Papertype	An integer value; 0=don't change, 1=A4, 2=US Letter, 3=US Legal, 4=B5.
Preview	An integer value; set to 0; this is not used. Under version 7, it is used as a placeholder for legacy FaxPack applications.

Return Value:

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

Example:

\$Err:=SPIImageControl(Defaults;ScalingPercent;StartPgNum;EndPgNum;Grayscale;PaperType;Preview)

Server Control

SPSetLogin

Syntax: *SPSetLogin*(Username:S; Password:S):ResultCode:I

Purpose:

The SPSetLogin command provides a required mechanism to establish a valid User name and password in order for other FaxPack commands to log-in to perform necessary actions or retrieve server-based information. This command is required and should be issued prior to using the other FaxPack commands.

Parameters:

Name	Explanation
UserName	A string expression that provides the username of the User to be logged in.
Password	A string expression that provides the password of this User.

Return Value:

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

Example:

\$Err:=SPSetLogin(Username;Password)

SPCheckLogin

Syntax: *SPCheckLogin*:ResultCode:I

Purpose:

The SPCheckLogin command is an optional command that validates the User name and password previously used within the SPSetLogin command. This command ensures the log-in information is legitimate allowing users to proceed with certainty. No parameters are required as the username and password values from the SPSetLogin command are used. Before using this command, issue the SPSetLogin after the SPSetServer (or SPPickServer) command.

Parameters:

None

Return Value:

Name	Explanation
ResultCode	If the operation is successful, a value of zero is returned. If an invalid User or Password is entered, a 10052 error code is returned. If another error occurs, a non-zero error code value is returned.

Example:

\$Err:=SPCheckLogin

SPPickServer

Syntax: *SPPickServer(ServerName:S; ServerType:S; Server IP Address:S; Server Port:L;Timeout:I;MaxToFind:I):ResultCode:I*

Purpose:

The SPPickServer command allows you to display a dialog to enable the user to select a valid FAX-Server. This is required when the application is first installed or the FAX server configuration is changed. After the user selects a server, pending and future messages are sent to the selected server when SPEndMessage is called.

The command will return a description of the chosen server in the parameters passed so that they may be saved by your application and restored (via SPSetServer) for subsequent sessions. Within each TCP/IP network, it may take several seconds search time to locate and display 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 4D Developer can increase the "TMEOUT" time to a larger number.

CAUTION: 4D Developers are not encouraged to use large numbers, because the "Pick Server" dialog may wait until this number of seconds has transpired before appearing. 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).

Parameters:

Name	Explanation
ServerName	A string variable that will be assigned the name of the selected server.
ServerType	This parameter is no longer used in

	version 7, yet it provides support for legacy FaxPack applications.
Server IP Address	A string variable that will be assigned the TCP/IP Address of the 4-Sight FAX Server.
Server Port	A long integer variable that will be assigned the Port Number of the 4-Sight FAX Server.
Timeout	An integer variable that specifies the maximum number of seconds to search before presenting the list of available fax servers (on the current local area network (LAN)). If 0 (seconds) is entered, 5 is used.
MaxToFind	An integer variable that specifies the maximum number of FAX servers to find before terminating the search.

Return Value:

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

Example:

\$Err:=SPPickServer(ServerName;ServerType;ServerZone;ServerAddr;ServerPort;Timeout;MaxToFind):

SPGetServer

Syntax: *SPGetServer(ServerName:S; ServerType:S; Server IP Address:S; Server Port:L); ResultCode:I*

Purpose:

The SPGetServer command returns the values from the last SPPickServer or SPSetServer call.

Parameters:

Name	Explanation
ServerName	A string variable that will be assigned the name of the currently selected server.
ServerType	A string variable that is not used, but is provided to support legacy FaxPack applications.
Server IP Address	A string variable that will be assigned the TCP/IP Address of the 4-Sight FAX Server
Server Port	A long integer variable that will be assigned the Port Number of the 4-Sight FAX Server

Return Value:

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

Example:

```
$Err:=SPGetServer(ServerName;ServerType;ServerIPAddress;ServerPort)
```

SPSetServer

Syntax: *SPSetServer(ServerName:S; ServerType:S; Server IP Address:S; Server Port:L; AllowPicker:I):ResultCode:I*

Purpose:

The SPSetServer command is used to identify or change the server for pending and future messages in the current application session. The server description is obtained by calling the SPPickServer or SPGetServer commands.

If the specified server is valid and available, then the connection is established and messages will be sent to the specified server when the SPEndMessage command is called.

If the specified server is not available and the value passed for the AllowPicker parameter is (1), the command will display the server-picker dialog to allow the user to select a server. If the user selects a server, messages will be sent to the selected server when the SPEndMessage command is called and a description of the selected server will be returned as values of the

first four parameters. Messages cannot be sent to a server until a valid server has been identified.

Parameters:

Name	Explanation
ServerName	A string variable that contains the name of the desired server. If Allow- Picker is true and the user selects a server, this variable will be assigned the name of the selected server.
ServerType	This is not used in version 7. It provides support legacy to older FaxPack applications.
Server IP Address	A string variable that will be assigned the TCP/IP Address of the 4- Sight FAX Server.
Server Port	A long integer variable that will be assigned the Port Number of the 4-Sight FAX Server.
AllowPicker	An integer expression; if this parameter is 1 and the values of the preceding parameters do not identify a valid server, the Server Picker dialog box will be displayed to allow the user to select a new, valid server. Assuming the user selects a Server, the description of the selected server will be returned as values of the preceding parameters.

Return Value:

Name	Explanation
ResultCode	<p>If the operation is successful, a value of zero is returned. If an error occurs, a non-zero error code value is returned.</p> <p>If the specified server is not available and the value passed for the AllowPicker parameter is (0), the command will return the value (10108).</p> <p>If the user cancels the server-picker dialog, the command will return a (-128) value.</p>

Example:

\$Err:=SPSetServer(ServerName;ServerType;ServerIPAddress;ServerPort)`Last parameter is omitted

SPGetCovers

Syntax: *SPGetCovers(NamesArray:X): ResultCode:I*

Purpose:

The SPGetCovers command allows you to retrieve a list of the cover page names defined in the current server. A maximum of 14 cover pages are supported. Prior to issuing the SPGetCovers command, the 4D Application must code the following:

Array Text(namesarray;0)

Parameters:

Name	Explanation
NamesArray	<p>A text array that is used to return the list of cover page names. After the call has completed, the array will contain an element for each cover page, whose value will be the name of that page. This array must be declared using 4D's ARRAY TEXT command; any data stored in the array prior to the call will be lost.</p>

Return Value:

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

Example:

```
Array Text(NamesArray;0)
```

```
$Err:= SPGetCovers(NamesArray)
```

SPGetServerVers

Syntax: *SPGetServerVers(Version:l): ResultCode:l*

Purpose:

The SPGetServerVers command allows you to determine the version number of the current 4-Sight FAX server. This command is provided to test for features which may require a specific server version. You must have assigned a server for the current session using SPSetServer or SPPickServer before calling this command.

Parameters:

Name	Explanation
Version	An integer value, with the format 0xMMmb where MM is the major version number, m is the minor version number and b is the bug- fix release number. The version 7.0.4 would be represented by the number 0x0704.

Return Value:

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

Example:

```
$Err:= SPGetServerVers(Version)
```

SPNotify

Syntax: *SPNotify(NotifyFlag:!) : ResultCode:!*

Purpose:

The SPNotify command is not currently functional in FaxPack version 7. It remains available for legacy FaxPack applications requiring this command, however. The syntax is as follows.

Parameters:

Name	Explanation
NotifyFlag	An integer value. Formerly used to enable 4-Sight FAX client notification. If it was enabled, the developer would pass a value of 1. If notification was disabled, the developer would pass a 0.

Return Value:

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

Example:

\$Err:= SPNotify (1)

Queue Access

SPGetQueueListing

Syntax: *SPGetQueueListing(Listcode:I;ItemID:X;Date:X;Time:X;From:X;FromNumber:X;To:X;Subject:X;State:X;Status:X;NumberofPages:X;Size:X;Position:X;Commtype:X): ResultCode:I*

Purpose:

The SPGetQueueListing command downloads a queue listing from the FAX Server. The queue listing provides an array element for each data field within a queue. All of the values within a specific element comprise all of the data available for each fax. The parameter descriptions are as follows.

Parameters:

Name	Explanation
List Code	An integer value. 1 = all faxes out, 2 = all faxes in, 3 = my faxes out, 4 = my faxes in
ItemID	A long integer array. The unique ID for the facsimile.
Date	A Date Array. The date of the facsimile.
Time	A Time Array. The time of the facsimile.
From	A text array. The person who sent the facsimile.
FromNumber	A text array. The fax number the facsimile came from
To	A text array. The recipient of the facsimile.
Subject	A text array. The subject of the facsimile.
State	An integer array. The state of the facsimile . 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 text array. The current status of the facsimile where the following possible values exist as of v7.0: "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.
Number of Pages	An integer array. The number of pages for the facsimile.
Size	A long integer array. The physical size, in bytes, of the facsimile.
Position	A long integer array. This refers to the position # within the outgoing queue.
CommType	An integer array where 1 = fax and 2 = email.

Return Value:

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

Example:

SPGetQueueListing(\$Listcode;ItemIDs;FaxDates;FaxTimes;FaxFroms;FromNumbers;FaxTos; FaxSubjects;FaxStates;FaxStatus;NumberofPages;Sizes;Positions;CommTypes)

\$state:= FaxStates{1} ` get state of 1st element

\$wasSent:=((\$state & 0x04) # 0)

\$wasRejected:=((\$state & 0x08) # 0)

SPGetFaxTiff

Syntax: *SPGetFaxTiff(Queue:I; ItemID:L;Part:I;Attachment Number:I;Tiff File Path;S):ResultCode:I*

Purpose:

The SPGetFaxTiff allows the 4D application to retrieve the fax file from the 4-Sight FAX server. This command is useful for Tiff File retrieval for databases providing facsimile viewing capabilities. Developers can retrieve either a one page preview (Part = 1) (for the first page) and attachments for outgoing faxes and they retrieve either the one page preview or the attachment (i.e., the entire fax).

Parameters:

Name	Explanation
Queue	An integer value. 1 = outgoing, 2 = incoming
ItemID	A long integer. The unique ID for the facsimile.
Part	An integer value. 1 = One Page Preview (valid for incoming and outgoing), 2 = cover page (valid for outgoing only), 3 = attachment (valid for incoming and outgoing).
Attachment #	An integer expression Used if "Part" is equal to "3".
Tiff File Path	A string expression providing the full path name of the Tiff file location.

Return Value:

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

Example:

\$Err:=SPGetFaxTiff(QueueType;ItemID;Part;AttachmentNumber;TiffFilePath)

SPArchiveItem

Syntax: *SPArchiveItem (Queue:I; ItemID:L;ArchiveKind:I):ResultCode:I*

Purpose:

The SPArchiveItem command sends a request to the 4-Sight FAX server to archive the given queue item. The queue item must be marked "completed" for it to be archived. For incoming faxes "completed" means the fax has been read and if printing is turned on it must have been printed. For outgoing faxes "completed" means the fax has been sent and if printing is turned on it must have been printed. If a queue item is part of a broadcast fax then it will not be archived until all related recipients have been sent to.

Parameters:

Name	Explanation
Queue	An integer value. 1 = outgoing, 2 = incoming
ItemID	A long integer. The unique ID for the facsimile.
ArchiveKind	An integer value. 1 = Original format, 2 = PDF File, 3 = Both. (PDF is only available for incoming faxes).

Return Value:

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

Example:

\$Err:= SPArchiveItem (QueueType;ItemID;ArchiveKind)

Image Control

SPGetTiffInfo

Syntax: *SPGetTiffInfo(TiffFilePath:S;NumberofPages:I):ResultCode:I*

Purpose

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

Parameters:

Name	Explanation
Tiff File Path	An string value specifying the Tiff source file's full path name.
Number of Pages	An integer indicating the number of pages in the file specified in Tiff File Path.

Return Value:

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

Example:

```
$Err:=SPGetTiffInfo(TiffFilePath;NumberofPages)
```

SPConvertTiff

```
Syntax: SPConvertTiff(ConvertType:I;TiffFilePath:S;DestFilePath:S;StartPageNum:I;EndPageNum:I): ResultCode:I
```

Purpose:

The SPConvertTiff 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 SPConvertTiff command, the developer should call the SPGetFaxTiff command to first retrieve the Tiff file to be converted.

Parameters:

Name	Explanation
Convert Type	An integer value specifying the file type, where 1=PDF, 2=Pict, 3 = PNG, 4 = BMP

	and 5 = Tiff.
Tiff File Path	A string(255) value containing the full path filename of the Fax Tiff file.
Dest File Path	A string(255) value containing the full path filename of the saved file name.
Start Page Number	An integer indicating the starting page number to be saved (from the file specified in Tiff File Path).
End Page Number	An integer indicating the ending page number to be saved (from the file specified in Tiff File Path).

Return Value:

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

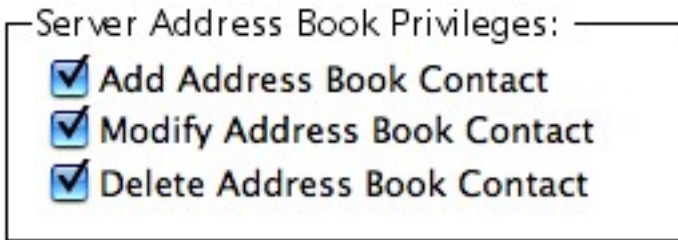
Example:

`$Err:= SPConvertTiff (3;$MyFaxFile;$NewFileName;$StartPageNum;$EndPageNum)`

Address Book

Server Setup:

Please contact your 4-Sight FAX Administrator to ensure those FaxPack Users who intend to work with the 4-Sight FAX Address Book are setup 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.



SPGetContact

Syntax: *SPGetContact(ContactID:L;Contactfields:X):ResultCode:I*

Purpose:

The SPGetContact command returns the contact record information from the 4-Sight FAX Server address book into a text (or string) array based on the ContactID value provided.

Parameters:

Name	Explanation
ContactID	A long integer value specifying the ID of the Contact desired.
Contactfields	A text (or string) array for all of the Contact fields available within the Address Book. {1} ContactID {2} Full Name {3} First Name, {4} Last Name, {5} Salutation {6} Title, {7} Company {8} Category, {9} Smart Dialing ("0" = no, "1" = yes) {10} Country Code {11} Area Code {12} Fax Number {13} Email Address {14} Phone Number {15} Address1 {16} Address2, {17} City, {18} State/Province {19} Country, {20} Postal Code {21} Custom 1, {22} Custom 2 & {23} Custom 3.

Return Value:

Name	Explanation
ResultCode	An integer value. If the operation is successful, a value of zero is returned. If a ContactID of 0 is passed, the error code returned is 10023. If the ContactID is not 0 and is not found, an error code of 10090 is returned. When the contact record is currently being edited by someone else, a 10091 error code is returned. When the user does not have this privilege (as defined on the FAX Server), a 10057 error code is returned. If other errors occur, a non-zero error code value is returned.

Example:

`$Err:= SPGetContact (ContactID;Contactfields)`

SPModifyContact

Syntax: `SPModifyContact(ContactID:L;Contactfields:X):ResultCode:I`

Purpose:

The SPModifyContact command updates the contact address record information from the 4-Sight FAX Server address book with the data in the populated ContactFields array. The developer must provide the ContactID for the desired Contact from the 4-Sight FAX Server Address Book.

Parameters:

Name	Explanation
ContactID	A long integer value specifying the ID of the Contact desired.
Contactfields	A text (or string) array for all of the Contact fields available within the Address Book. {1} ContactID {2} Full Name {3} First Name, {4} Last Name, {5} Salutation {6} Title, {7} Company {8} Category, {9} Smart Dialing ("0" = no, "1" = yes) {10} Country Code {11} Area Code {12} Fax Number

	{13} Email Address {14} Phone Number {15} Address1 {16} Address2, {17} City, {18} State/Province {19} Country, {20} Postal Code {21} Custom 1, {22} Custom 2 & {23} Custom 3
--	--

Return Value:

Name	Explanation
ResultCode	An integer value. If the operation is successful, a value of zero is returned. If a ContactID of 0 is passed, the error code returned is 10023. If the ContactID is not 0 and is not found, an error code of 10090 is returned. When the contact record is being edited by someone else, a 10091 error code is returned. When the user does not have this privilege (as defined on the FAX Server), a 10057 error code is returned. If other errors occur, a non-zero error code value is returned.

Example:

\$Err:= SPMModifyContact (ContactID;Contactfields)

SPAddContact

Syntax: *SPAddContact (ContactID:L;Contactfields:X):ResultCode:I*

Purpose:

The SPAddContact command adds the contact address record information to the 4-Sight FAX Server address book with the data in the populated ContactFields array. The ContactID value will be created by the 4-Sight FAX Server and returned into both the ContactID parameter as well as within the first array element.

Parameters:

Name	Explanation
ContactID	A long integer value where the ID of the new Contact record is returned.
Contactfields	A text (or string) array for all of the Contact fields available within the Address Book. {1} ContactID (serves as redundancy for the ContactID field) {2} Full Name {3} First Name {4} Last Name {5} Salutation {6} Title {7} Company {8} Category, {9} Smart Dialing ("0" = no, "1" = yes) {10} Country Code {11} Area Code {12} Fax Number {13} Email Address {14} Phone Number {15} Address1 {16} Address2 {17} City {18} State/Province {19} Country, {20} Postal Code {21} Custom 1, {22} Custom 2 & {23} Custom 3

Return Value:

Name	Explanation
ResultCode	An integer value. If the operation is successful, a value of zero is returned. When the user does not have this privilege (as defined on the FAX Server), a 10057 error code is returned. If other errors occur, a non-zero error code value is returned.

Example:

\$Err:= SPAddContact (ContactID;Contactfields)

SPDeleteContact

Syntax: *SPDeleteContact(ContactID:L):ResultCode:I*

Purpose:

The SPDeleteContact command deletes the contact address record from the 4-Sight FAX Server address book with the given ContactID field.

Parameters:

Name	Explanation
ContactID	A long integer value where the ID of the Contact record to be deleted is assigned.

Return Value:

Name	Explanation
ResultCode	An integer value. If the operation is successful, a value of zero is returned. If a ContactID of 0 is passed, the error code returned is 10023. If the ContactID is not 0 and is not found, an error code of 10090 is returned. When the contact record is being edited by someone else, a 10091 error code is returned. When the user does not have this privilege (as defined on the FAX Server), a 10057 error code is returned. If other errors occur, a non-zero error code value is returned.

Example:

```
$Err:= SPDeleteContact (ContactID)
```

SPGetAddressBook

Syntax: *SPGetAddressBook(Dest File Path:S;Columns:X;Encoding:I):ResultCode:I*

Purpose:

The SPGetAddressBook command returns the entire 4-Sight FAX Server address book into a tab-delimited text file (default). Alternatively, developers requiring a Unicode compatible format

can select the UTF8 format. For the "Columns" parameter: each array element contains a contact field number. A column in the text file will be created for each array element.

Parameters:

Name	Explanation
DestFilePath	A string variable specifying the full path to the file you wish to create. If you wish to allow the user to specify the location of the new file, assign a null string to this variable before calling the command; in this case, the command will display a standard file package "Save As" dialog to allow the user to enter a file name and select the folder in which to save the file. When the command has completed, the full path to the file chosen by the user will be assigned to this variable.
Columns	An integer array where each element is a Contact field number in the Address Book. {1} ContactID (serves as redundancy for the ContactID field) {2} Full Name {3} First Name {4} Last Name {5} Salutation {6} Title {7} Company {8} Category, {9} Smart Dialing ("0" = no, "1" = yes) {10} Country Code {11} Area Code {12} Fax Number {13} Email Address {14} Phone Number {15} Address1 {16} Address2 {17} City {18} State/Province {19} Country, {20} Postal Code {21} Custom 1, {22} Custom 2 & {23} Custom 3.
Encoding Kind	0 = Mac Roman, 1 = UTF8. Mac Roman is the default format. UTF8 is Unicode encoding format.

Return Value:

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

Example:

ARRAY INTEGER(columns;5)

columns{1}:=1 ` Contact ID

columns{2}:=4 ` Last Name

columns{3}:=3 ` First Name

columns{4}:=7 ` Company

columns{5}:=12 ` Fax Number

\$err:=SPGetAddressBook(\$destFilePath;columns;0)

Will create a tab-delimited text file with 5 columns:

1st column - Contact IDs

2nd column - Last Names

3rd column - First Names

4th column - Companies

5th column - Fax Numbers

Block List

SPAddToBlockList

Syntax: *SPAddToBlockList(Name:S;Number:S):ResultCode:I*

Purpose:

The SPAddToBlockList command adds the name of the fax sender and the fax number to the 4-Sight FAX Server Block List. By doing this, future incoming faxes from this fax number will be halted at the time of recognition.

Parameters:

Name	Explanation
Name	Name of the Company that is sending faxes that you want blocked.
Fax Number	The Fax Number you are receiving unwanted faxes from that you want blocked.

Return Value:

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

Example:

```
$Err:= SPAddToBlockList ($Name;$ FaxNumber)
```

Server Setup:



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

SPDeleteFromBlockList

Syntax: *SPDeleteFromBlockList(Number:S):ResultCode:I*

Purpose:

The SPDeleteFromBlockList 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.

Parameters:

Name	Explanation
Fax Number	The Fax Number you were previously blocking faxes from that you no longer want blocked.

Return Value:

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

Example:

```
$Err:= SPDeleteFromBlockList ($FaxNumber)
```

Command Summary

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

Value	Definition
SPRegister	SPRegister (DeveloperName:S; Mac RegNumber:S;Win RegNumber:S;Deployment Lic#:S): ResultCode:S
SPGetVersion	SPGetVersion : ResultCode:
SPStartMessage	SPStartMessage (From:S; Subject:S; CoverPage:L; Reserved:I): ResultCode:I
SPAddAddress	SPAddAddress (Name:S; Company:S; Fax Number:S; IsUrgent:I; DestKind:I; SendDate:D; Send- Time:H; Country Code:S; Area Code:S; Other Fields:X): ResultCode:I
SPAddComment	SPAddComment (Comment:T; Resolution:L): ResultCode:I
SPEndMessage	SPEndMessage (Message ID:S; SingleUser:I; AllowPicker:I): ResultCode:I
SPSaveMessage	SPSaveMessage (FilePath:S; SFPrompt:S; SFDefault:S): ResultCode:I
SPSaveMessage	SPSaveMessage (FilePath:S; SFPrompt:S; SFDefault:S): ResultCode:I
SPAbandonMsg	SPAbandonMsg
SPGetItemStatus	SPGetItemStatus (UniqueMessage ID:S; QueueTypeOrIndex:I; PageCount:I; Status:S; From:S; To:S; Subject:S; Address:S; Number:S; NextSendTime:H; IsDone:I; IsPrinted:I; IsRejected:I;IsDeleted:I): ResultCode:I
SPAddEnclosure	SPAddEnclosure (Filename:S): ResultCode:I
SPPickEnclosure	SPPickEnclosure (Filename:S): ResultCode:I
SPPickMany	SPPickMany (DefaultPath:S;FilesArray:X): ResultCode:I
SPGetFilePath	SPGetFilePath (Filename:S): ResultCode:I
SPSetPrinter	SPSetPrinter (DeviceName:S): ResultCode:I
SPSwitchPrinter	SPSwitchPrinter (Mode:I): ResultCode:I
SPPreparePrint	SPPreparePrint (Resolution:L): ResultCode:I
SPGetPrinter	SPGetPrinter (DeviceName:S): ResultCode:I

SPImageControl	SPImageControl (Defaults:I;Scaling Percent:I;Starting Page Number:I;Ending Page Number:I;Grayscale:I;PaperType:I;Preview:I): ResultCode:L
SPSetLogin	SPSetLogin (Username:S; Password:S):ResultCode:I
SPCheckLogin	SPCheckLogin :ResultCode:I
SPPickServer	SPPickServer (ServerName:S; ServerType:S; Server IP Address:S; Server Port:L;Timeout:I;MaxToFind:I):ResultCode:I
SPGetCovers	SPGetCovers (NamesArray:X): ResultCode:I
SPGetServerVers	SPGetServerVers (Version:I): ResultCode:I
SPNotify	SPNotify (NotifyFlag:I): ResultCode:I
SPGetQueueListing	SPGetQueueListing (Listcode:I;ItemID:X;Date:X;Time:X;From:X;FromNumber:X; To:X;Subject:X;State:X;Status:X;NumberofPages:X;Size:X;Position:X;Commtype:X): ResultCode:I
SPGetFaxTiff	SPGetFaxTiff (Queue:I; ItemID:L;Part:I;Attachment Number:I;Tiff File Path;S):ResultCode:I
SPGetFaxTiff	SPGetFaxTiff (Queue:I; ItemID:L;Part:I;Attachment Number:I;Tiff File Path;S):ResultCode:I
SPGetTiffInfo	SPGetTiffInfo (TiffFilePath:S;NumberofPages:I):ResultCode:I
SPConvertTiff	SPConvertTiff (ConvertType:I;TiffFilePath:S;DestFilePath:S;StartPageNum:I; EndPageNum:I): ResultCode:I
SPGetContact	SPGetContact (ContactID:L;Contactfields:X):ResultCode:I
SPModifyContact	SPModifyContact (ContactID:L;Contactfields:X):ResultCode:I
SPAddContact	SPAddContact (ContactID:L;Contactfields:X):ResultCode:I
SPDeleteContact	SPDeleteContact (ContactID:L):ResultCode:I
SPGetAddressBook	SPGetAddressBook (Dest File Path:S;Columns:X;Encoding:I): ResultCode:I
SPAddToBlockList	SPAddToBlockList (Name:S;Number:S):ResultCode:I
SPDeleteFromBlockList	SPDeleteFromBlockList (Number:S):ResultCode:I

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.
-43	File Not Found (Either it was deleted from Fax Server or the program has incorrectly specified the file identification).
-128	Dialogue cancelled by user.
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.
10012	4-Sight Fax printer driver version 7.0 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 zero). New in version 7.
10024	This feature is not supported by the Fax Server.
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. New in v7.
10073	"From" name is different than the User name, and the "Allow user name alias" is unchecked at the server. New in version 7.
10082	Unknown archive kind. New in 7.5.
10083	Archive kind not available for this queue item. New in 7.5.
10084	Cannot archive; the incoming fax has not been read. New in 7.5.
10085	Cannot archive; the incoming fax has not been printed. New in 7.5.
10086	Cannot archive; the outgoing fax has not been sent. New in 7.5.
10087	Cannot archive; not all related recipients have been sent to. New in 7.5.
10088	Cannot archive; the outgoing fax has not been printed. New in 7.5.
10089	The queue item with the given ID does not exist. New in 7.5.
10090	the Contact with the given ID was not found. New in version 7.5.
10091	the Contact is being edited by another user. New in version 7.
10093	The PDF archive could not be created New in version 7.5.
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.
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	"SPStartMessage" 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 "SPGetPrinter" & "SPSwitchPrinter".
10214	There were no calls to "SPPreparePrint".
10215	There was more than one call to "SPPreparePrint".
10217	The fax number parameter to "SPAddAddress" was empty.
10218	Invalid conversion type.

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 SPswitchprinter command, go to the print queue and not the fax queue?

Be sure to always include the SPPreparePrint command (prior to issuing the 4D Print command), as this command is necessary for proper faxing. This will allow the SPswitchPrinter command to work properly. Also, ensure the SPStartMessage and SPSendMessage 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 SPSetServer 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.

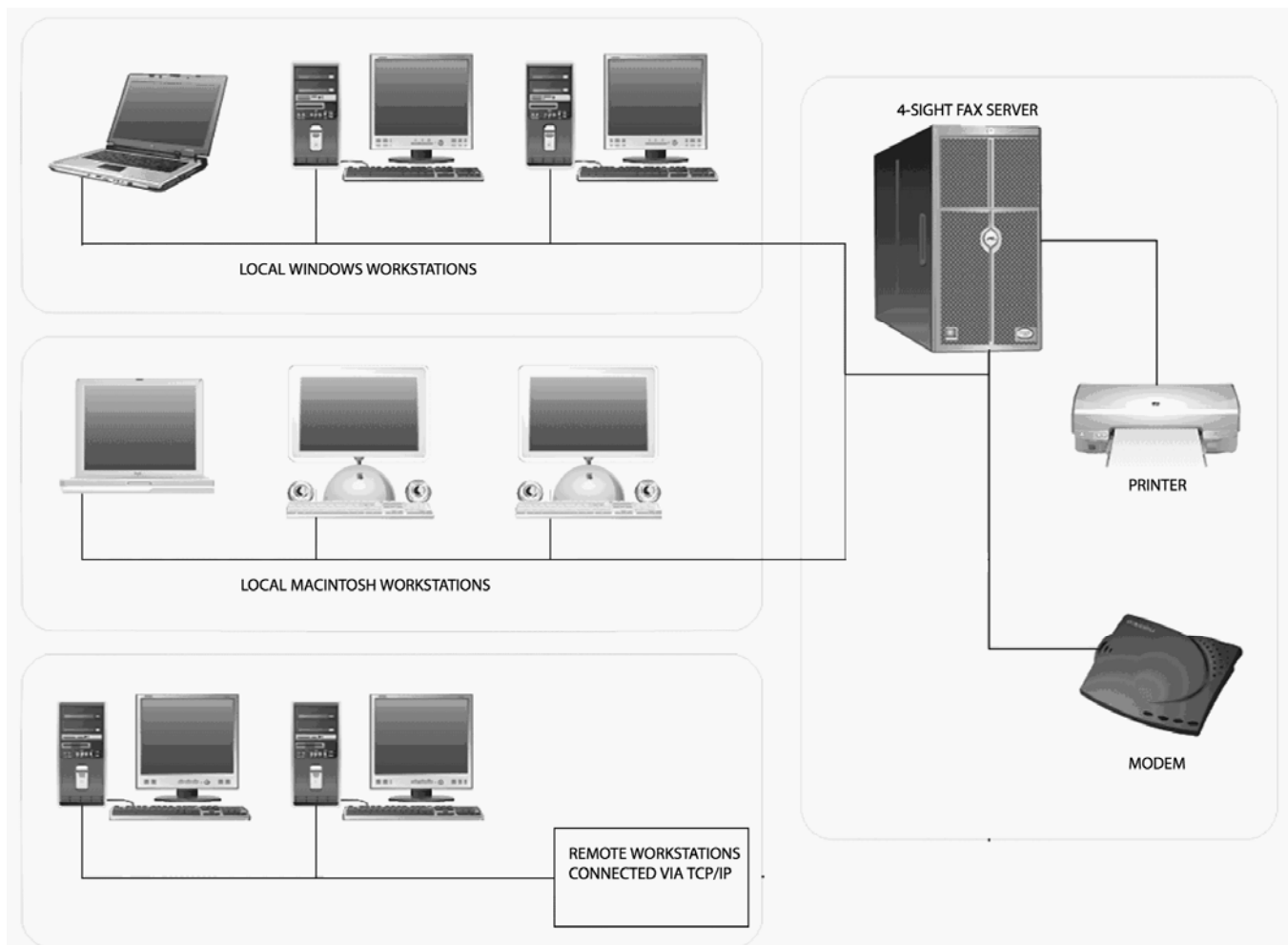
Why do I receive a 10250 error?

This is the error returned for "Demo Expired".

In the plug-in, FaxPack captures the current TIME from the plug-in on 4D startup. If the 4D programmer correctly issues our registration command, we no longer make the TIME comparison, because the developer is registered and has unrestricted access to using the plug-in. If the 4D developer has not issued a valid FaxPack registration, we allow the product to run in demonstration mode for 45 Minutes.

If you experience this error message, please ensure your registration information is correct.

4-Sight FAX Sample Configuration



Contact Information

Customer Service or Sales: 770-457-9400 or mail to: sales@softsinc.com

Technical Support: 770-457-9400 or mail to: support@softsinc.com

For the latest versions of FaxPack for 4D, please visit:
<http://www.4sightfax.com/faxpack4d.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

Convert Type, 47
FromNumber, 43, 44, 59
Grayscale, 33, 34, 59
Installation, 7, 8
NotifyFlag, 42, 59
Papertype, 34
Preview, 33, 34, 45, 59
Scaling Percent, 33, 59
Server IP Address, 36, 37, 38, 39, 59
Server Port, 36, 37, 38, 39, 59
ServerName, 36, 37, 38, 39, 40, 59
ServerType, 36, 37, 38, 39, 40, 59
SPAbandonMsg, 5, 6, 13, 22, 58
SPAddContact, 6, 52, 59
SPAddEnclosure, 4, 5, 26, 28, 58
SPAddToBlockList, 56, 59
SPCheckLogin, 3, 35, 36
SPConvertTiff, 5, 47, 48, 59
SPDeleteContact, 53, 59
SPDeleteFromBlockList, 57, 59
SPEndMessage, 3, 5, 13, 16, 19, 20, 23, 26, 36, 38, 58
SPGetAddressBook, 6, 53
SPGetContact, 5, 49, 50, 53, 59
SPGetFaxTiff, 5, 45, 46, 47, 59
SPGetFilePath, 4, 28, 29, 58
SPGetItemStatus, 5, 20, 23, 25, 58
SPGetPrinter, 3, 32, 33, 58, 62
SPGetQueueListing, 23, 43, 44, 59
SPGetServerVers, 6, 41, 59
SPGetTiffInfo, 5, 46, 47, 59
SPGetVersion, 6, 11, 13, 58
SPImageControl, 33, 34, 59
SPModifyContact, 6, 50, 51, 59
SPNotify, 6, 42, 59
SPPickEnclosure, 5, 21, 26, 27, 58
SPPickMany, 5, 21, 27, 28, 58
SPPreparePrint, 3, 4, 31, 32, 58, 62, 63
SPRegister, 3, 11, 12, 13, 58
SPSaveMessage, 4, 5, 20, 22, 58
SPSetLogin, 3, 35, 59
SPSetPrinter, 3, 30, 31, 58
SPSetServer, 3, 35, 36, 38, 40, 41
SPSwitchPrinter, 3, 9, 30, 31, 58, 62, 63
Timeout, 36, 37, 59