

# 4-Sight Fax API v1.3 for 4-Sight Fax 7 Reference

August 26, 2008

## **SP\_Init**

---

Initializes the API.

```
SP_Err SP_Init(void);
```

function result      If a value of SP\_NO\_ERROR is returned the initialization was successful.

### DESCRIPTION

The SP\_Init function must be called before any other API function can be called. Call this function once in the initialization phase of your application or plug-in.

## **SP\_Deinit**

---

Releases resources used by the API.

```
SP_Err SP_Deinit(void);
```

function result      If a value of SP\_NO\_ERROR will always be returned.

### DESCRIPTION

Call this function once in the de-initialization phase of your application or plug-in.

## **SP\_CreateStringRefFromCString**

---

Create a string reference from a null-terminated string.

```
SP_Err SP_CreateStringRefFromCString(const char *inString_s, SP_StringRef *outStringRef);
```

inString\_s            On input, a pointer to a null-terminated string.

outStringRef         A pointer to a 'SP\_StringRef' variable. On output, if successful, a string reference. If an error occurred, or 'inString\_s' has zero length, then 'outStringRef' will be set to SP\_NULL\_REF.

function result      A value of SP\_NO\_ERROR if no errors were encountered creating the string reference.

## DESCRIPTION

Creates a string reference from a null-terminated string. A null-terminated string is just a 'C' string such as "Hello World". Once the string reference is no longer needed, release the reference with "SP\_ReleaseRef".

## **SP\_CreateStringRefFromUnicode**

---

Create a string reference from Unicode characters.

```
SP_Err SP_CreateStringRefFromUnicode (SP_UInt4B inNumChars,  
                                     const SP_UniChar *inChars_p,  
                                     SP_StringRef *outStringRef);
```

inNumChars      On input, the number of Unicode characters.

inChars      On input, a pointer to the Unicode characters.

outStringRef      A pointer to a 'SP\_StringRef' variable. On output, if successful, a string reference. If an error occurred, or 'inNumChars' is zero, then 'outStringRef' will be set to SP\_NULL\_REF.

function result      A value of SP\_NO\_ERROR if no errors were encountered creating the string reference.

## DESCRIPTION

Create a string reference from Unicode characters. Once the string reference is no longer needed, release the reference with "SP\_ReleaseRef".

## **SP\_GetCStringFromStringRef**

---

Get a 'C' string from a string reference.

```
SP_Err SP_GetCStringFromStringRef (SP_StringRef inStringRef,  
                                   char *outString_s,  
                                   SP_UInt4B inMaxLen);
```

inStringRef      On input, a string reference. SP\_NULL\_REF can be passed producing a string with just the null-terminated byte.

outString\_s      A pointer to where the null-terminated string characters are to be stored. On output, the characters ending with a null-terminated byte.

inMaxLen	On input, the maximum length the string can hold not including the null-terminated byte.
function result	A value of SP_NO_ERROR if no errors were encountered producing the characters.

## DESCRIPTION

Gets the characters out of a string reference and places them into a null-terminated byte string.

## **SP\_GetUnicodeFromStringRef**

---

Get Unicode characters from a string reference.

```
SP_Err SP_GetUnicodeFromStringRef (SP_StringRef inStringRef,
                                   SP_UInt4B *outNumChars,
                                   SP_UniChar *outChars_p,
                                   SP_UInt4B inMaxLen);
```

inStringRef	On input, a string reference. SP_NULL_REF can be passed and 'outChars_p' will return with a value of zero.
outNumChars	A pointer to a 'SP_UInt4B' variable. On output, the number of characters returned in 'outChars_p'.
outChars_p	A pointer to where Unicode characters are to be stored. On output, the Unicode characters.
inMaxLen	On input, the maximum number of Unicode characters that 'outChars_p' can hold.
function result	A value of SP_NO_ERROR if no errors were encountered producing the characters.

## DESCRIPTION

Gets the Unicode characters out of a string reference.

## **SP\_DateTimeToSeconds**

---

Convert a specific date & time to a 'SP\_DateTime' value.

```
SP_DateTime SP_DateTimeToSeconds(SP_UInt2B inDay,
                                  SP_UInt2B inMonth,
                                  SP_UInt2B inYear,
```

```
SP_UInt2B inHour,  
SP_UInt2B inMinute,  
SP_UInt2B inSecond);
```

inDay	On input, the day value of the date (1 thru 31).
inMonth	On input, the month value of the date (1 thru 12).
inYear	On input, the year value of the date (greater than or equal to 1970).
inHour	On input, the hour value of the time (0 thru 23).
inMinute	On input, the minute value of the time (0 thru 59).
inSecond	On input, the second value of the time (0 thru 59).
function result	A 'SP_DateTime' value.

## DESCRIPTION

Convert a specific date & time to a 'SP\_DateTime' value. A 'SP\_DateTime' value must be passed to the "SP\_StartMessage" call in the 'inSendDateTime' parameter.

## **SP\_SecondsToDateTime**

---

Gets the date & time out of a 'SP\_DateTime' value.

```
void SP_SecondsToDateTime      (SP_DateTime inDTSeconds,  
                                SP_UInt2B *outDay,  
                                SP_UInt2B *outMonth,  
                                SP_UInt2B *outYear,  
                                SP_UInt2B *outHour,  
                                SP_UInt2B *outMinute,  
                                SP_UInt2B *outSecond);
```

inDTSeconds	On input, a 'SP_DateTime' value.
outDay	A pointer to a 'SP_UInt2B' variable. On output, the day value of the date (1 thru 31).
outMonth	A pointer to a 'SP_UInt2B' variable. On output, the month value of the date (1 thru 12).
outYear	A pointer to a 'SP_UInt2B' variable. On output, the year value of the date (greater than or equal to 1970).
outHour	A pointer to a 'SP_UInt2B' variable. On output, the hour value of the time (0 thru 23).

outMinute	A pointer to a 'SP_UInt2B' variable. On output, the minute value of the time (0 thru 59).
outSecond	A pointer to a 'SP_UInt2B' variable. On output, the second value of the time (0 thru 59).

## DESCRIPTION

Gets the date & time out of a 'SP\_DateTime' value. A 'SP\_DateTime' value can be obtained with "SP\_DateTimeToSeconds".

## **SP\_CreateFileRefFromPath**

---

Create a file reference from a full file path.

```
SP_Err SP_CreateFileRefFromPath(SP_StringRef inFilePath, SP_FileRef *outFileRef);
```

inFilePath	On input, a string reference of a full file path. For example: "HD:Documents:Test". The file does not have to exist.
outFileRef	A pointer to a 'SP_FileRef' variable. On output, a file reference. If an error occurs then 'outFileRef' will be set to SP_NULL_REF.
function result	A value of SP_NO_ERROR if no errors were encountered creating the file reference.

## DESCRIPTION

Create a file reference from a full file path. The path is platform specific, so on a Macintosh the path might be "HD:Documents:Test" but on Windows it will be "C:\Documents\Test". The file does not have to exist. Once the file reference is no longer needed, release the reference with "SP\_ReleaseRef".

## **SP\_CreateFileRefFromFSSpec**

---

Create a file reference from a Macintosh FSSpec structure.

```
SP_Err SP_CreateFileRefFromFSSpec(const FSSpec *inFSSpec_p, SP_FileRef *outFileRef);
```

inFSSpec	A pointer to a FSSpec structure. On input, a valid FSSpec usually created with the toolbox "FSMakeFSSpec" call. The file does not have to exist.
outFileRef	A pointer to a 'SP_FileRef' variable. On output, a file reference. If an error occurs then 'outFileRef' will be set to SP_NULL_REF.

function result      A value of SP\_NO\_ERROR if no errors were encountered creating the file reference.

## DESCRIPTION

Create a file reference from a Macintosh FSSpec structure. The file does not have to exist. This call will only be available for Macintosh applications or plug-ins. Once the file reference is no longer needed, release the reference with "SP\_ReleaseRef".

## **SP\_RetainRef**

---

Increment a reference's count value.

SP\_Err SP\_RetainRef(SP\_Ref inRef);

inRef                      On input, a reference. Any reference type that was derived from 'SP\_Ref' will do, for example a 'SP\_StringRef' or 'SP\_FileRef'. A SP\_NULL\_REF value can be passed in here but nothing will happen of course.

function result      A value of SP\_NO\_ERROR if the increment was successful.

## DESCRIPTION

Increment a reference's count value. When a reference is created it will have its count set to 1. A reference and its resources will not be released until its count becomes zero with a call to "SP\_ReleaseRef".

## **SP\_ReleaseRef**

---

Decrement a reference's count value.

SP\_Err SP\_ReleaseRef(SP\_Ref inRef);

inRef                      On input, a reference. Any reference type that was derived from 'SP\_Ref' will do, for example a 'SP\_StringRef' or 'SP\_FileRef'. A SP\_NULL\_REF value can be passed in here but nothing will happen of course.

function result      A value of SP\_NO\_ERROR if the decrement was successful.

## DESCRIPTION

Decrement a reference's count value. If the count becomes zero the reference and its resources are released and the reference can no longer be used.

## **SP\_GetVersion**

---

Get the API's version.

```
SP_Err SP_GetVersion(char *outVersion_s);
```

outVersion\_s            A pointer to where the version string is to be stored. On output, the API version as a null-terminated byte string.

function result        A value of SP\_NO\_ERROR will always be returned.

### DESCRIPTION

Get the API's version as a null-terminated byte string.

## **SP\_GetUserName**

---

Get the name of the machine's owner.

```
SP_Err SP_GetUserName(SP_StringRef *outName);
```

outName                A pointer to a 'SP\_StringRef' variable. On output, the name of the machine's owner. On MacOS 8/9 this is the value in the "Owner Name" edit box on the File Sharing dialog window.

function result        A value of SP\_NO\_ERROR if no errors were encountered.

### DESCRIPTION

Get the name of the machine's owner.

## **SP\_GetFaxPrinterVersion**

---

Get the version of the 4-Sight Fax print driver.

```
SP_Err SP_GetFaxPrinterVersion(char *outVersion_s);
```

outName                A pointer to where the string is to be stored. On output, the current version of the 4-Sight Fax print driver, as a null-terminated byte string.

function result        A value of SP\_NO\_ERROR if no errors were encountered. A value of SPERR\_FAXPRINTER\_NOT\_FOUND if the fax printer does not exist.

### DESCRIPTION

Get the version of the 4-Sight Fax print driver. If the print driver does not exist the function will return a SPERR\_FAXPRINTER\_NOT\_FOUND error.

## **SP\_GetPrinter**

---

Get a reference to the current print driver.

```
SP_Err SP_GetPrinter(SP_PrinterRef *outPrinterRef);
```

outPrinterRef        A pointer to a 'SP\_PrinterRef' variable. On output, a print driver reference. If an error occurs then 'outPrinterRef' will be set to SP\_NULL\_REF.

function result        A value of SP\_NO\_ERROR if no errors were encountered.

### DESCRIPTION

This returns a reference to the current print driver. For example, "LaserWriter 8". The reference returned with this function is to be used to restore the current print driver with a "SP\_SetPrinter" call. Once the print driver reference is no longer needed, release the reference with "SP\_ReleaseRef".

## **SP\_SetPrinter**

---

Set the current print driver using the given print driver reference.

```
SP_Err SP_SetPrinter(SP_PrinterRef inPrinterRef);
```

inPrinterRef        On input, a print driver reference obtained with "SP\_GetPrinter".

function result        A value of SP\_NO\_ERROR if no errors were encountered.

### DESCRIPTION

Set the current print driver using the given print driver reference.

## **SP\_GetPrinterAsString**

---

Get a string representation of the current print driver.

```
SP_Err SP_GetPrinterAsString(SP_StringRef *outStringRef);
```

outStringRef            A pointer to a 'SP\_StringRef' variable. On output, a string representation of the current print driver. If an error occurs then 'outStringRef' will be set to SP\_NULL\_REF.

function result        A value of SP\_NO\_ERROR if no errors were encountered.

## DESCRIPTION

This returns a string representation of the current print driver. The string reference returned with this function can be used to restore the current print driver with a "SP\_SetPrinterFromString" call.

Getting the current print driver as a string representation fulfills the need to store the string and use it to restore the current print driver at a later date.

## **SP\_SetPrinterFromString**

---

Set the current print driver using the given string representation.

```
SP_Err SP_SetPrinterFromString(SP_StringRef inStringRef);
```

inStringRef            On input, a string reference. The value of the string should have initially come from a call to "SP\_GetPrinterAsString".

function result        A value of SP\_NO\_ERROR if no errors were encountered.

## DESCRIPTION

Set the current print driver using the given string representation. The value of the string should have initially come from a call to "SP\_GetPrinterAsString" at an earlier date.

## **SP\_GetPrinterName**

---

Get the name of the current print driver.

```
SP_Err SP_GetPrinterName(SP_StringRef *outNameRef);
```

outNameRef            A pointer to a 'SP\_StringRef' variable. On output, the name of the current print driver. If an error occurs then 'outStringRef' will be set to SP\_NULL\_REF.

function result        A value of SP\_NO\_ERROR if no errors were encountered.

## DESCRIPTION

Get the name of the current print driver.

## **SP\_SetToFaxPrinter**

---

Set the current print driver to the 4-Sight Fax print driver.

```
SP_Err SP_SetToFaxPrinter(void);
```

function result        A value of SP\_NO\_ERROR if no errors were encountered.

### DESCRIPTION

Set the current print driver to the 4-Sight Fax print driver. In order to capture printing output to a TIFF file, the current print driver must be the 4-Sight Fax print driver.

## **SP\_PreparePrintCapture**

---

Prepare to capture printing output.

```
SP_Err SP_PreparePrintCapture(SP_FileRef inDestFileRef, SP_PrintRef *outPrintRef);
```

inDestFileRef        On input, a file reference to a file which is intended to be the destination of TIFF data created from the printing output of a print job. The file reference can be obtained with any SP\_CreateFileRefFromXXXX call.

outPrintRef        A pointer to a 'SP\_PrintRef' variable. On output, a print reference. If an error occurs then 'outPrintRef' will be set to SP\_NULL\_REF.

function result        A value of SP\_NO\_ERROR if no errors were encountered.

### DESCRIPTION

This is the first call in a series to capture printing output to a TIFF file. Note though that the current print driver should be the 4-Sight Fax print driver before this call. This can be assured by using the "SP\_SetToFaxPrinter" call beforehand. "SP\_PreparePrintCapture" must be called before printing is started. After printing, either "SP\_WaitCaptureComplete" or "SP\_AbandonCapture" should be called. Once the print reference is no longer needed, release the reference with "SP\_ReleaseRef".

## **SP\_SetCaptureSettings**

---

Set the settings for capturing print output.

```
SP_Err SP_SetCaptureSettings (SP_PrintRef inPrintRef,  
                             SP_UInt2B inPaperType,  
                             SP_UInt2B inScaleFactor,  
                             SP_Bool inLargePrintArea,  
                             SP_UInt2B inStartPage,
```

SP\_UInt2B inEndPage,  
SP\_UInt2B inResolutionKind,  
SP\_Bool inGrayScale);

inPrintRef	On input, a print reference obtained from "SP_PreparePrintCapture".
inPaperType	On input, the paper type. Values are: 0 = use default 1 = A4 2 = US Letter 3 = US Legal 4 = B5 Constants, prefixed with 'SP_PAPER_TYPE_', have been defined for the above values.
inScaleFactor	On input, the reduce/enlarge percentage. Values range from 25 to 500. A value of 100 is normal.
inLargePrintArea	On input, a flag to use a Larger Print Area. Use the constants 'SP_TRUE' or 'SP_FALSE'.
inStartPage	On input, the page number in which to begin printing. All pages will be printed if this value is zero AND the value for 'inEndPage' is zero.
inEndPage	On input, the page number in which to end printing. All pages will be printed if this value is zero AND the value for 'inStartPage' is zero.
inResolutionKind	On input, the resolution kind. Use the constants 'SP_RESOLUTION_STANDARD' for standard resolution, or 'SP_RESOLUTION_FINE' for fine resolution.
inGrayScale	On input, a flag to use a gray scale. Use the constants 'SP_TRUE' for gray scale, or 'SP_FALSE' for black & white.
function result	A value of SP_NO_ERROR if no errors were encountered.

## DESCRIPTION

Set the page setup & print settings for capturing print output. This is an optional function. If used, it is the second call in a series to capture printing output to a TIFF file. It must be called before a print job is started. If it is not used then the settings will use defaults.

## **SP\_WaitCaptureComplete**

---

Wait for printing output to be converted to a TIFF file.

SP\_Err SP\_WaitCaptureComplete (SP\_PrintRef inPrintRef,  
SP\_UInt4B inTimeoutSecs,  
SP\_RefCon inRefCon,

SPCallback inCallback);

inPrintRef	On input, a print reference obtained from "SP_PreparePrintCapture".
inTimeoutSecs	On input, the number of seconds to wait for the TIFF file to be complete. A value of zero indicates no limit.
inRefCon	On input, a value for developer's use. If a callback is used, this value will be passed to the callback function in the 'ioInfo_p->mRefCon' field.
inCallback	On input, a pointer to a function of type 'SPCallback'. A callback is not required so a 'NULL' value can be passed.
function result	A value of SP_NO_ERROR if no errors were encountered.

## DESCRIPTION

This is the final call in a series to capture printing output to a TIFF file. It must be called after printing. After an application prints, the TIFF file is not necessarily complete. This function will not return until the TIFF file is complete, the timeout value has been reached, or the callback signals a cancel. If a callback is used, it will be called periodically with the 'ioInfo\_p->mRefCon' field set to the value passed in 'inRefCon'. You can cancel the "SP\_WaitCaptureComplete" call from the callback by setting the 'ioInfo\_p->mStatus' field to the constant 'SP\_USER\_CANCEL\_ERROR'. This value will also be returned in the function result.

## **SP\_AbandonCapture**

---

Cancel a print capture initiated by "SP\_PreparePrintCapture".

SP\_Err SP\_AbandonCapture(SP\_PrintRef inPrintRef);

inPrintRef	On input, a print reference obtained from "SP_PreparePrintCapture".
------------	---

function result	A value of SP_NO_ERROR if no errors were encountered.
-----------------	---

## DESCRIPTION

Cancel a print capture initiated by "SP\_PreparePrintCapture". Use this function instead of "SP\_WaitCaptureComplete" if you would to cancel the print capture process. "SP\_ReleaseRef" should be called afterwards with 'inPrintRef'.

## **SP\_ServerLookup**

---

Finds 4-Sight Fax Servers on the local network.

SP\_Err SP\_ServerLookup(SP\_UInt4B inTimeoutSecs,

SP\_RefCon inRefCon,  
SPServerLookupCallBack inCallBack);

inTimeoutSecs      On input, the number of seconds to wait for Server discovery. A value of zero indicates no limit.

inRefCon            On input, a value for developer's use. This value will be passed to the callback function in the 'ioInfo\_p->mRefCon' field.

inCallBack          On input, a pointer to a function of type 'SPServerLookupCallBack'.

function result     A value of SP\_NO\_ERROR if no errors were encountered.

## DESCRIPTION

Finds 4-Sight Fax Servers on the local network. A callback function is required. The callback function will be called for each Server found. The callback will be handed a 'SP\_ServerLookupInfo' structure. The fields are as follows:

mRefCon      same value as 'inRefCon'.  
mIPAddress   the TCP/IP address of the Server.  
mPort        the TCP/IP port number in which the Server listens (usually 8008).  
mName        the name of the Server machine.  
mDone        the callback can set to 'SP\_TRUE' if it wants the lookup to stop.

When the callback returns, the API will call "SP\_ReleaseRef" on the 'mIPAddress' and 'mName' fields. So, the callback should call "SP\_RetainRef" if it wants to keep either reference.

## **SP\_RetrieveServerInfo**

---

Retrieves information about a 4-Sight Fax Server.

```
SP_Err SP_RetrieveServerInfo(SP_StringRef inServerIPAddress,  
                             SP_UInt2B inServerPort,  
                             char *outAPIVersion_s,  
                             char *outServerVersion_s,  
                             SP_UInt2B *outPlatform,  
                             char *outPlatformVersion_s,  
                             SP_UInt2B *outCompatibility);
```

inServerIPAddress    On input, the TCP/IP address of the Server.

inServerPort        On input, the TCP/IP port number of the Server.

outAPIVersion\_s     A pointer to where the string is to be stored. On output, the current version of the Client/Server API, as a null-terminated byte string.

outServerVersion\_s   A pointer to where the string is to be stored. On output, the version of the Server, as a null-terminated byte string.

outPlatform	A pointer to a 'SP_UInt2B' variable. On output, the OS on which the Server is running. Values are (1 = Classic MacOS, 2 = MacOS X).
outPlatformVersion_s	A pointer to where the string is to be stored. On output, the version of OS on which the Server is running, as a null-terminated byte string.
outCompatibility	A pointer to a 'SP_UInt2B' variable. On output, the compatibility of this API with the Server. Values are: 1 = Compatible 2 = Incompatible; not licensed for the API's platform 3 = Incompatible; this API is older than what the Server supports 4 = Incompatible; this API is newer than what the Server supports
function result	A value of SP_NO_ERROR if no errors were encountered.

## DESCRIPTION

Retrieves information about a 4-Sight Fax Server. If the function returns with SP\_NO\_ERROR then the 'outCompatibility' value should be checked for compatibility with this API.

## **SP\_CreateLoginRef**

---

Creates a login reference to be used to communicate with a Server.

```
SP_Err SP_CreateLoginRef (SP_LoginRef *outLoginRef,
                        SP_StringRef inServerIPAddress,
                        SP_UInt2B inServerPort,
                        SP_StringRef inUserName,
                        SP_StringRef inPassword);
```

outLoginRef	A pointer to a 'SP_LoginRef' variable. On output, a login reference. If an error occurred 'outLoginRef' will be set to SP_NULL_REF.
inServerIPAddress	On input, the TCP/IP address of the Server.
inServerPort	On input, the TCP/IP port number of the Server.
inUserName	On input, the name of the User to login as.
inPassword	On input, the password of the User. A SP_NULL_REF can be passed which is interpreted as an empty password.
function result	A value of SP_NO_ERROR if no errors were encountered.

## DESCRIPTION

Creates a login reference to be used with several other calls to send or receive information from a Server. Note: This call does not attempt to connect to the Server, and as such cannot verify if the UserName and/or Password is valid on the Server.

Once the login reference is no longer needed, release the reference with "SP\_ReleaseRef".

## **SP\_CheckLogin**

---

Attempts to connect to a Server and checks if the UserName/Password can login.

```
SP_Err SP_CheckLogin(SP_LoginRef inLoginRef);
```

inLoginRef            On input, a login reference obtained from a call to "SP\_CreateLoginRef".

function result        A value of SP\_NO\_ERROR if no errors were encountered.

### DESCRIPTION

Attempts to connect to a Server and login with the UserName/Password. If the login was successful the call will logout and return a 'SP\_NO\_ERROR' result.

## **SP\_StartMessage**

---

Starts the building of an outgoing fax.

```
SP_Err SP_StartMessage(SP_MsgRef *outMsgRef,  
                      SP_FileRef inDestFileRef,  
                      SP_StringRef inFrom,  
                      SP_StringRef inSubject,  
                      SP_DateTime inSendDateTime,  
                      SP_UInt2B inPriority,  
                      SP_UInt2B inCoverPage,  
                      SP_UInt2B inCoverPageResKind,  
                      SP_StringRef inComment,  
                      SP_Bool inIsPrivate);
```

outMsgRef            A pointer to a 'SP\_MsgRef' variable. On output, a message reference. If an error occurred 'outMsgRef' will be set to SP\_NULL\_REF.

inDestFileRef        On input, a file reference to a file in which the building of the outgoing fax will be stored. The file will be created if it does not exist.

inFrom                On input, the value for the 'From' field.

inSubject            On input, the value for the 'Subject' field.

inSendDateTime	On input, the date & time when the fax should be sent. The constant 'SP_DATETIME_NOW' can be passed if the fax is to be sent as soon as possible. Otherwise a 'SP_DateTime' value can be obtained with a call to "SP_DateTimeToSeconds".
inPriority	On input, the fax send priority. Value are: 0 = Bulk 25 = Normal 50 = ASAP 75 = Important 100 = Urgent Constants, prefixed with 'SP_PRIORITY_', have been defined for the above values.
inCoverPage	On input, the number of the cover page to use. Pass a value of zero for no cover page.
inCoverPageResKind	On input, the resolution kind to use for imaging the cover page. Use the constants 'SP_RESOLUTION_STANDARD' for standard resolution, or 'SP_RESOLUTION_FINE' for fine resolution.
inComment	On input, the value for the 'Comment' field. Pass SP_NULL_REF for no comment.
inIsPrivate	On input, a flag to indicate a private fax. Use the constants 'SP_TRUE' for private, or 'SP_FALSE' for non-private.
function result	A value of SP_NO_ERROR if no errors were encountered.

## DESCRIPTION

This is the first in a series of calls to store information into a file for an outgoing fax. Once the file is built it can be sent to a Server, sent to the Composer, or just stored for later use. Use "SP\_AddRecipient" to add recipients, and "SP\_AddTIFF" to add TIFF attachments. Use either "SP\_FinishMessage" to complete the file, or "SP\_AbandonMessage" to abort. Once the message reference is no longer needed, release the reference with "SP\_ReleaseRef".

## **SP\_AddRecipient**

---

Adds a recipient to the building of an outgoing fax.

```
SP_Err SP_AddRecipient (SP_MsgRef inMsgRef,
                        SP_UInt2B inNumFields,
                        SP_StringRef inFieldVals[ ]);
```

inMsgRef            On input, a message reference obtained from "SP\_StartMessage".

inNumFields	On input, the number of recipient fields in 'inFieldVals'. The number can be less than the constant 'NUM_RECIPIENT_FIELDS' but should not exceed it.
inFieldVals	<p>On input, an array of recipient field values. These are the index values into the array:</p> <ul style="list-style-type: none"> <li>0 = Full Name</li> <li>1 = Salutation</li> <li>2 = First Name</li> <li>3 = Last Name</li> <li>4 = Smart Dial ( "1" = yes, "0" or SP_NULL_REF = no )</li> <li>5 = Country Code (used only for Smart Dial)</li> <li>6 = Area Code (used only for Smart Dial)</li> <li>7 = Fax Number (entire number if not Smart Dial)</li> <li>8 = Company</li> <li>9 = Title</li> <li>10 = Category</li> <li>11 = Alternate</li> <li>12 = Voice</li> <li>13 = Email</li> <li>14 = Address line 1</li> <li>15 = Address line 2</li> <li>16 = City</li> <li>17 = Province/State</li> <li>18 = Country</li> <li>19 = Postal Code</li> <li>20 = Custom 1</li> <li>21 = Custom 2</li> <li>22 = Custom 3</li> </ul> <p>Constants, prefixed with 'RCPNTFIELD_', have been defined for the above values. "Full Name" and "Fax Number" are required. All other fields can be empty, so 'SP_NULL_REF' can be used.</p>
function result	A value of SP_NO_ERROR if no errors were encountered.

## DESCRIPTION

Adds a recipient to the building of an outgoing fax. At least one recipient must be added for the outgoing fax to be valid. To indicate that Smart Dial is to be used, create a string reference with the string "1", otherwise a string "0". If Smart Dial is on, then Country Code, Area Code, and Fax Number are used to determine the actual number dialed. If Smart Dial is off, Country Code and Area Code are ignored and the number dialed will be Fax Number.

## **SP\_AddTIFF**

---

Adds a TIFF file attachment to the building of an outgoing fax.

SP\_Err SP\_AddTIFF(SP\_MsgRef inMsgRef, SP\_FileRef inTIFF\_FileRef);

inMsgRef	On input, a message reference obtained from "SP_StartMessage".
inTIFF_FileRef	On input, a file reference to the TIFF file to be attached.
function result	A value of SP_NO_ERROR if no errors were encountered.

## DESCRIPTION

Adds a TIFF file attachment to the building of an outgoing fax. The data of the TIFF file is not added but only a link. When the outgoing fax is sent to the Server the link is used to open, read and send the TIFF file's data to the Server.

## **SP\_FinishMessage**

---

Completes the building of an outgoing fax.

```
SP_Err SP_FinishMessage(SP_MsgRef inMsgRef);
```

inMsgRef	On input, a message reference obtained from "SP_StartMessage".
function result	A value of SP_NO_ERROR if no errors were encountered.

## DESCRIPTION

Completes the building of an outgoing fax. The file passed to "SP\_StartMessage" in the 'inDestFileRef' parameter is now complete. The file can now be passed to "SP\_SendMessage", "SP\_SendMessageToComposer", or just kept for later use. A call to "SP\_ReleaseRef" with 'inMsgRef' should follow after "SP\_FinishMessage".

## **SP\_AbandonMessage**

---

Aborts the building of an outgoing fax.

```
SP_Err SP_AbandonMessage(SP_MsgRef inMsgRef);
```

inMsgRef	On input, a message reference obtained from "SP_StartMessage".
function result	A value of SP_NO_ERROR if no errors were encountered.

## DESCRIPTION

Aborts the building of an outgoing fax. The file passed to "SP\_StartMessage" in the 'inDestFileRef' parameter is not deleted though. If desired, an OS should be called to delete it. A call to "SP\_ReleaseRef" with 'inMsgRef' should follow after "SP\_AbandonMessage".

## SP\_SendMessage

---

Sends the outgoing fax data contained in a file to a Server.

```
SP_Err SP_SendMessage (SP_FileRef inMsgFileRef,  
                      SP_LoginRef inLoginRef,  
                      SP_MsgID *outMsgID,  
                      SP_RefCon inRefCon,  
                      SPCallBack inCallBack);
```

inMsgFileRef	On input, a file reference to a file containing outgoing fax information. The file is usually created with a series of "building an outgoing fax" calls that begins with "SP_StartMessage".
inLoginRef	On input, a login reference obtained with "SP_CreateLoginRef". The fax information will be sent to the Server and under the User contained in this login reference.
outMsgID	A pointer to a 'SP_MsgID' variable. On output, this will contain a unique ID which the Server maintains and has passed back. This value can be used in the "SP_RetrieveFaxStatus" call.
inRefCon	On input, a value for developer's use. This value will be passed to the callback function in the 'ioInfo_p->mRefCon' field.
inCallBack	On input, a pointer to a function of type 'SPCallBack'. A callback is not required so a 'NULL' value can be passed.
function result	A value of SP_NO_ERROR if no errors were encountered.

### DESCRIPTION

Sends the outgoing fax data contained in a file to a Server. If a callback is used it will be handed a 'SP\_CallbackInfo' structure. The fields are as follows:

mRefCon	same value as 'inRefCon'.
mTotalBytes	the total number of bytes to be transferred.
mBytesTransferred	the number of bytes transferred so far.
mStatus	the callback can set to 'SP_USER_CANCEL_ERROR' to cancel sending.
mWhat	what state the sending process is in. Values are:

- 1 = Connecting to Server
- 2 = Logging In
- 3 = Logging Out
- 4 = Issuing a command
- 5 = Sending Data
- 6 = Receiving Data
- 7 = Sending an attachment

Constants, prefixed with 'SP\_WHAT\_', have been defined for the above values.

If successful, the function will return with SP\_NO\_ERROR and 'outMsgID' will contain an ID which can be used in "SP\_RetrieveFaxStatus".

## **SP\_SendMessageToComposer**

---

Launches the Composer and sends it a message to use the outgoing fax data contained in this file.

```
SP_Err SP_SendMessageToComposer(SP_FileRef inMsgFileRef);
```

**inMsgFileRef**            On input, a file reference to a file containing outgoing fax information. The file is usually created with a series of "building an outgoing fax" calls that begins with "SP\_StartMessage".

**function result**        A value of SP\_NO\_ERROR if no errors were encountered.

### DESCRIPTION

Launches the Composer and sends it a message to use the outgoing fax data contained in this file.

## **SP\_RetrieveFaxStatus**

---

Retrieves information about a fax that was sent with "SP\_SendMessage".

```
SP_Err SP_RetrieveFaxStatus(SP_LoginRef inLoginRef,  
                              const SP_MsgID* inMsgID,  
                              SP_UInt4B inIndex,  
                              SP_ItemDetailInfo *outItemDetail);
```

**inLoginRef**            On input, a login reference obtained with "SP\_CreateLoginRef".

**inMsgID**                A pointer to a 'SP\_MsgID' variable. On input, the message ID value that was obtained in the 'outMsgID' parameter to a "SP\_SendMessage" call.

**inIndex**                On input, the index of the recipient. A value of 1 is the first recipient. The index of the recipient matches the order in which they were placed in the outgoing fax with "SP\_AddRecipient".

**outItemDetail**        A pointer to a 'SP\_ItemDetailInfo' variable. On output, information about the outgoing fax. The fields are as follows:

<b>mItemID</b>	unique ID of the item.
<b>mFlags</b>	status flags. A combination of these values:
	1 Sent
	2 Printed
	4 Rejected
	8 Deleted

	Constants, prefixed with 'STATUS_', have been defined for the above values.
mPageCount	number of pages.
mReserved	reserved for future use.
mHasCoverPage	a value of 'SP_TRUE' if it has a cover page, 'SP_FALSE' if it does not have a cover page.
mAttachmentCount	number of attachments.
mSendDateTime	a 'SP_DateTime' value of the sending, or sent, date & time.
mFrom	from
mTo	recipient's full name.
mSubject	subject
mFaxNumber	fax number to be dialed.
mStatus	status text.
mReason	if the fax was rejected, the reason.

function result      A value of SP\_NO\_ERROR if no errors were encountered.

## DESCRIPTION

Retrieves information about a fax that was sent with "SP\_SendMessage".

## **SP\_RetrieveCoverPagesInfo**

---

Retrieves the names of the cover pages on a given Server.

```
SP_Err SP_RetrieveCoverPagesInfo (SP_LoginRef inLoginRef,
                                   SP_UInt2B inMaxCount,
                                   SP_UInt2B *outActualCount,
                                   SP_StringRef outNames[ ]);
```

inLoginRef      On input, a login reference obtained with "SP\_CreateLoginRef".

inMaxCount      On input, the maximum number of elements 'outNames' can hold.

outActualCount    A pointer to a 'SP\_UInt2B' variable. On output, the actual number of cover page names returned.

outNames      A pointer to an array of 'SP\_StringRef'. On output, the cover page names.

function result    A value of SP\_NO\_ERROR if no errors were encountered.

## DESCRIPTION

Retrieves the names of the cover pages on a given Server.

## SP\_RetrieveFax

---

Downloads the image of an incoming or outgoing fax to a local file.

SP\_Err SP\_RetrieveFax (SP\_LoginRef inLoginRef,  
SP\_UInt2B inQueue,  
SP\_ItemID inItemID,  
SP\_UInt2B inPart,  
SP\_UInt2B inAttachmentNum,  
SP\_FileRef inDestFileRef,  
SP\_RefCon inRefCon,  
SP\_Callback inCallBack);

inLoginRef	On input, a login reference obtained with "SP_CreateLoginRef".
inQueue	On input, a value of 'SP_QUEUE_OUTGOING' or 'SP_QUEUE_INCOMING'.
inItemID	On input, the queue item ID of the fax.
inPart	On input, a value of one of the following: 'SP_PART_PREVIEW' – the first page of the fax. 'SP_PART_COVERPAGE' – the cover page. 'SP_PART_ATTACHMENT' – one of the attachments.
inAttachmentNum	On input, the attachment number if 'inPart' is 'SP_PART_ATTACHMENT'.
inDestFileRef	On input, a file reference to a local file which is to hold the image. The file will be a TIFF file.
inRefCon	On input, a value for developer's use. This value will be passed to the callback function in the 'ioInfo_p->mRefCon' field.
inCallBack	A pointer to a function of type 'SP_Callback' (can be NULL).
function result	A value of SP_NO_ERROR if no errors were encountered.

### DESCRIPTION

Downloads the image of an incoming or outgoing fax to a local file. The file will be a TIFF file. For incoming faxes use 'SP\_PART\_ATTACHMENT' for the 'inPart' parameter, and a value of 1 for 'inAttachmentNum'.

## SP\_RetrieveQueueListing

---

Downloads a listing of the incoming or outgoing queue to a local file.

SP\_Err SP\_RetrieveQueueListing (SP\_LoginRef inLoginRef,  
SP\_UInt2B inQueue,  
SP\_UInt2B inAllOrMine,  
SP\_FileRef inDestFileRef,  
SP\_RefCon inRefCon,  
SP\_Callback inCallBack);

inLoginRef	On input, a login reference obtained with "SP_CreateLoginRef".
inQueue	On input, a value of 'SP_QUEUE_OUTGOING' or 'SP_QUEUE_INCOMING'.
inAllOrMine	On input, a value of 1 is all queue items, otherwise only my queue items.
inDestFileRef	On input, a file reference to a local file which is to hold the listing.
inRefCon	On input, a value for developer's use. This value will be passed to the callback function in the 'ioInfo_p->mRefCon' field.
inCallBack	A pointer to a function of type 'SP_Callback' (can be NULL).
function result	A value of SP_NO_ERROR if no errors were encountered.

### DESCRIPTION

Downloads a listing of the incoming or outgoing queue to a local file. The file will be in tab-delimited format.

## SP\_GetListingInfo

---

Gets information on a queue listing that was downloaded to a local file.

SP\_Err SP\_GetListingInfo (SP\_FileRef inListingFileRef,  
SP\_UInt4B \*outNumItems,  
SP\_UInt2B inMaxNumCols,  
SP\_UInt2B \*outActualNumCols,  
SP\_UInt2B outCols[]);

inListingFileRef	On input, a file reference to a local file which has the queue listing.
outNumItems	A pointer to a 'SP_UInt4B' variable. On output, the number of queue items in the listing.
inMaxNumCols	On input, the maximum number of columns that 'outCols' can hold.

outActualNumCols	A pointer to a 'SP_UInt2B' variable. On output, the actual number of columns returned in 'outCols'.
outCols	A pointer to an array of 'SP_UInt2B'. On output, the column types. The enum for column types all begin with the prefix "SPCOL_" (such as 'SPCOL_SUBJECT').
function result	A value of SP_NO_ERROR if no errors were encountered.

## DESCRIPTION

Gets information on a queue listing that was downloaded to a local file.

## **SP\_GetListingItems**

---

Gets the values from columns in a queue listing that was downloaded to a local file.

SP_Err	SP_GetListingItems	(SP_FileRef inListingFileRef, SP_UInt2B inNumCols, SP_UInt2B inCols[], SP_UInt4B inStartRow, SP_UInt4B inNumRows, SP_StringRef outValues[]);
inListingFileRef		On input, a file reference to a local file which has the queue listing.
inNumCols		On input, the number of columns in 'inCols'.
inCols		A pointer to an array of 'SP_UInt2B'. On input, the column types to get values on. The enum for column types all begin with the prefix "SPCOL_" (such as 'SPCOL_SUBJECT').
inStartRow		On input, the row number to start with.
inNumRows		On input, the number of rows to get information on.
outValues		A pointer to an array of 'SP_StringRef'. On output, the values in the columns. 'outValues' must be allocated to be able to hold (inNumCols * inNumRows) items.
function result		A value of SP_NO_ERROR if no errors were encountered.

## DESCRIPTION

Gets the values from columns in a queue listing that was downloaded to a local file.

## **SP\_GetTIFFInfo**

---

Gets the number of pages in a TIFF file.

SP\_Err SP\_GetTIFFInfo (SP\_FileRef inTIFFFileRef,  
SP\_UInt2B \*outNumPages);

inTIFFFileRef On input, a file reference to a local file which has the queue listing.

outNumPages A pointer to a 'SP\_UInt2B' variable. On output, the number of pages in the TIFF file.

function result A value of SP\_NO\_ERROR if no errors were encountered.

### DESCRIPTION

Gets the number of pages in a TIFF file.

## **SP\_TIFF\_To\_TIFF**

---

Copies pages of a TIFF file to a new TIFF file.

SP\_Err SP\_TIFF\_To\_TIFF (SP\_FileRef inSrcTIFFFileRef,  
SP\_FileRef inDstTIFFFileRef,  
SP\_UInt2B inStartPage,  
SP\_UInt2B inEndPage);

inTIFFFileRef On input, a source file reference to a local TIFF file.

inDstTIFFFileRef On input, a destination file reference to a local file.

inStartPage On input, the starting page number to start the copy.

inEndPage On input, the ending page number to end the copy.

function result A value of SP\_NO\_ERROR if no errors were encountered.

### DESCRIPTION

Copies pages of a TIFF file to a new TIFF file.

## **SP\_TIFF\_To\_PICT** (*Macintosh Only*)

---

Copies a page of a TIFF file to a new PICT file.

SP_Err	SP_TIFF_To_PICT	(SP_FileRef inTIFFFileRef, SP_FileRef inPICTFileRef, SP_UInt2B inPageNum, SP_UInt2B inBitsPerPixel, SP_UInt2B inHorzRes, SP_UInt2B inVertRes);
inTIFFFileRef		On input, a source file reference to a local TIFF file.
inPICTFileRef		On input, a destination file reference to a local file.
inPageNum		On input, the page number to copy.
inBitsPerPixel		On input, the number of bits per pixels (values are 1,2,4,8,16,32).
inHorzRes		On input, the number of pixels per inch for horizontal resolution.
inVertRes		On input, the number of pixels per inch for vertical resolution.
function result		A value of SP_NO_ERROR if no errors were encountered.

### DESCRIPTION

Copies a page of a TIFF file to a new PICT file.

## **SP\_TIFF\_To\_PDF**

---

Copies pages of a TIFF file to a new PDF file.

SP_Err	SP_TIFF_To_PDF	(SP_FileRef inTIFFFileRef, SP_FileRef inPDFFileRef, SP_UInt2B inStartPage, SP_UInt2B inEndPage);
inTIFFFileRef		On input, a source file reference to a local TIFF file.
inPDFFileRef		On input, a destination file reference to a local file.
inStartPage		On input, the starting page number to start the copy.
inEndPage		On input, the ending page number to end the copy.

function result            A value of SP\_NO\_ERROR if no errors were encountered.

## DESCRIPTION

Copies pages of a TIFF file to a new PDF file.

## **SP\_TIFF\_To\_PNG**

---

Copies a page of a TIFF file to a new PNG file.

SP\_Err SP\_TIFF\_To\_PNG            (SP\_FileRef inTIFFFileRef,  
   SP\_FileRef inPNGFileRef,  
   SP\_UInt2B inPageNum);

inTIFFFileRef            On input, a source file reference to a local TIFF file.

inPNGFileRef            On input, a destination file reference to a local file.

inPageNum            On input, the page number to copy.

function result            A value of SP\_NO\_ERROR if no errors were encountered.

## DESCRIPTION

Copies a page of a TIFF file to a new PNG file.

## **SP\_TIFF\_To\_BMP**

---

Copies a page of a TIFF file to a new BMP file.

SP\_Err SP\_TIFF\_To\_BMP            (SP\_FileRef inTIFFFileRef,  
   SP\_FileRef inBMPFileRef,  
   SP\_UInt2B inPageNum,  
   SP\_UInt2B inBitsPerPixel,  
   SP\_UInt2B inHorzRes,  
   SP\_UInt2B inVertRes);

inTIFFFileRef            On input, a source file reference to a local TIFF file.

inBMPFileRef            On input, a destination file reference to a local file.

inPageNum            On input, the page number to copy.

inBitsPerPixel	On input, the number of bits per pixels (values are 1,2,4,8,16,32).
inHorzRes	On input, the number of pixels per inch for horizontal resolution.
inVertRes	On input, the number of pixels per inch for vertical resolution.
function result	A value of SP_NO_ERROR if no errors were encountered.

## DESCRIPTION

Copies a page of a TIFF file to a new BMP file.

## **SP\_RetrieveAddressBookEntry**

---

Retrieves a contact's information from the server address book.

```
SP_Err SP_RetrieveAddressBookEntry (SP_LoginRef inLoginRef,
                                     SP_UInt4B inContactID,
                                     SP_UInt2B inNumFields,
                                     const SP_UInt2B inFields[],
                                     SP_StringRef outFieldValues[]);
```

inLoginRef	On input, a login reference obtained with "SP_CreateLoginRef".
inContactID	On input, the ID number of the contact.
inNumFields	On input, the number of fields in 'inFields'.
inFields	A pointer to an array of 'SP_UInt2B'. On input, the field types to get values on. The enum for field types all begin with the prefix "CONTACTFIELD_" (such as 'CONTACTFIELD_FIRST_NAME').
outFieldValues	A pointer to an array of 'SP_StringRef'. On output, the values of the fields. 'outFieldValues' must be allocated to be able to hold 'inNumFields' items.
function result	A value of SP_NO_ERROR if no errors were encountered.

## DESCRIPTION

Retrieves a contact's information from the server address book.

## **SP\_ModifyAddressBookEntry**

---

Modify a contact's information in the server address book.

```
SP_Err SP_ModifyAddressBookEntry (SP_LoginRef inLoginRef,  
    SP_UInt4B inContactID,  
    SP_UInt2B inNumFields,  
    const SP_UInt2B inFields[],  
    const SP_StringRef inFieldValues[]);
```

inLoginRef            On input, a login reference obtained with "SP\_CreateLoginRef".

inContactID           On input, the ID number of the contact.

inNumFields           On input, the number of fields for 'inFields' & ' inFieldValues'.

inFields              A pointer to an array of 'SP\_UInt2B'. On input, the field types to set values on. The enum for field types all begin with the prefix "CONTACTFIELD\_" (such as 'CONTACTFIELD\_FIRST\_NAME').

inFieldValues         A pointer to an array of 'SP\_StringRef'. On input, the values of the fields.

function result       A value of SP\_NO\_ERROR if no errors were encountered.

## DESCRIPTION

Modify a contact's information in the server address book.

## **SP\_AddAddressBookEntry**

---

Add a contact to the server address book.

```
SP_Err SP_AddAddressBookEntry (SP_LoginRef inLoginRef,  
    SP_UInt2B inNumFields,  
    const SP_UInt2B inFields[],  
    const SP_StringRef inFieldValues[],  
    SP_UInt4B *outContactID);
```

inLoginRef            On input, a login reference obtained with "SP\_CreateLoginRef".

inNumFields           On input, the number of fields for 'inFields' & ' inFieldValues'.

inFields              A pointer to an array of 'SP\_UInt2B'. On input, the field types to set values on. The enum for field types all begin with the prefix "CONTACTFIELD\_" (such as 'CONTACTFIELD\_FIRST\_NAME').

inFieldValues         A pointer to an array of 'SP\_StringRef'. On input, the values of the fields.

outContactID          A pointer to a 'SP\_UInt4B' variable. On output, the ID number of the contact.



---

Delete a phone number from the server's CallerID block list.

SP\_Err SP\_DeleteFromBlockList (SP\_LoginRef inLoginRef,  
SP\_StringRef inNumber);

inLoginRef On input, a login reference obtained with "SP\_CreateLoginRef".

inNumber On input, the phone number to delete from the list.

function result A value of SP\_NO\_ERROR if no errors were encountered.

## DESCRIPTION

Delete a phone number from the server's CallerID block list.

---

## SP\_RetrieveAddressBook

---

Downloads a listing of the server's address book to a local file.

SP\_Err SP\_RetrieveAddressBook (SP\_LoginRef inLoginRef,  
SP\_FileRef inDestFileRef,  
SP\_UInt2B inEncodingKind,  
SP\_UInt2B inNumFields,  
SP\_UInt2B inFields [],  
SP\_RefCon inRefCon  
SPCallback inCallBack);

inLoginRef On input, a login reference obtained with "SP\_CreateLoginRef".

inDestFileRef On input, a file reference to a local file which is to hold the listing.

inEncodingKind On input, the type of encoding for the listing (values are 'SP\_ENCODING\_WCHAR', 'SP\_ENCODING\_UTF8', 'SP\_ENCODING\_MAC\_ROMAN').

inNumFields On input, the number of columns in 'inFields'.

inFields A pointer to an array of 'SP\_UInt2B'. . On input, the field types to retrieve a listing on. The enum for field types all begin with the prefix "CONTACTFIELD\_" (such as 'CONTACTFIELD\_FIRST\_NAME').

inRefCon On input, a value for developer's use. This value will be passed to the callback function in the 'ioInfo\_p->mRefCon' field.

inCallBack A pointer to a function of type 'SPCallback' (can be NULL).

function result A value of SP\_NO\_ERROR if no errors were encountered.

## DESCRIPTION

Downloads a listing of the server's address book to a local file.