SMTP/POP3/IMAP Email Engine

Library for Xbase++

Programmer's Manual

(SEE4XB)

Version 7.4

April 6, 2016

This software is provided as-is.
There are no warranties, expressed or implied.

MarshallSoft Computing, Inc.
Post Office Box 4543
Huntsville AL 35815

Email: info@marshallsoft.com
Web: www.marshallsoft.com

TABLE OF CONTENTS

1 Introduction

The **SMTP/POP3/IMAP Email Engine for Xbase++ (SEE4XB)** library is a toolkit that allows software developers to quickly develop SMTP and POP3/IMAP email applications in Alaska Xbase++.

The **SMTP/POP3/IMAP Email Engine (SEE)** is a component DLL library of functions that uses the Windows API to provide direct and simple control of the SMTP (Simple Mail Transport Protocol), POP3 (Post Office 3), and IMAP 4 (Internet Message Access Protocol) protocols.

A straightforward interface allows sending and receiving email, including multiple MIME base64 and quoted-printable encoded attachments, over any TCP/IP network (such as the Internet). Knowledge of Winsock and TCP/IP is not needed.

The **SMTP/POP3/IMAP Programmer's Manual** provides information needed to compile and run programs in an Alaska Xbase++ programming environment.

The **SMTP/POP3/IMAP Email Engine for Xbase++** component library supports and has been tested with Alaska Xbase++ version v1.3 through and Xbase++ v1.9. **SEE4XB** includes numerous example programs that demonstrate SMTP and POP3/IMAP functions used to create mail applications using the **SEE4XB** library.

SEE4XB runs under all 32-bit and 64-bit versions of Windows through Windows 10. The **SMTP/POP3/IMAP Email Engine SDK** DLLs (SEE32.DLL or SEE64.DLL) can also be used from any language (C/C++, .NET, Visual Basic, VB. NET, VBA, Delphi, Visual FoxPro, COBOL, PowerBASIC, dBASE, etc.) capable of calling the Windows API.

When comparing **SMTP/POP3/IMAP Email** component library against our competition, note that:

- SEE4XB is a standard Windows DLL and is much smaller than an OCX or ActiveX control.
- SEE4XB does NOT depend on ActiveX or similar "support" libraries.
- The WIN32 version of SEE is fully threadable.
- The SEE4XB functions can be called from applications not capable of using controls.

MarshallSoft also has versions of the **SMTP/POP3/IMAP Email Engine** library for C/C++ (SEE4C), Delphi (SEE4D), PowerBASIC (SEE4PB), Visual FoxPro (SEE4FP), Visual dBASE (SEE4DB), Visual Basic (SEE4VB) and COBOL (SEE4CB). All versions of the **SEE** library use the same DLL (SEE32.DLL or SEE64.DLL). However, the examples provided for each version are written for the specified programming environment.

The latest versions of **SMTP/POP3/IMAP Email Engine (SEE)** can be downloaded from our web site at

http://www.marshallsoft.com/email-component-library.htm

Our goal is to provide a robust SMTP/POP3/IMAP email component library that you and your customers can depend upon. A fully functional evaluation version is available. Contact us if you have any questions.

1.1 Features

Some of the many features of the **SMTP/POP3/IMAP Email Engine** component library are as follows:

- SMTP client for sending email.
- POP3/IMAPclient for receiving email.
- Send email with optional MIME or Quoted Printable attachments.
- Send email with inline embedded HTML, GIF, TIF, JPG, BMP and Rich Text attachments.
- Get the number of messages on the POP3/IMAP email server.
- Get the header lines from any email on the POP3/IMAP email server, without reading the entire email.
- Delete any email on the POP3/IMAP server without reading it first.
- Copy any email on the POP3/IMAP server without deleting it.
- Check for the number of emails on the POP3/IMAP server.
- Receive any email on the POP3/IMAP server including MIME attachments.
- Forward Email.
- Decoding email from a File
- Run up to 32 independent WIN32 threads concurrently.
- Can send email to mail addresses on a distribution list.
- Supports SMTP (ESMTP) and POP3 authentication.
- Set return receipt; add TO, CC, BCC recipients
- Set minimum and maximum wait times for server response.
- Supports ISO-8859 (European character sets) and UTF-8 (16 bit character sets) messages.
- Can specify custom Content-Types; add custom header fields
- Includes over 60 functions for SMTP and POP3/IMAP control.
- Dozens of switches provided to control how email is sent or received.
- Supports setting priority via X-Priority header field.
- Remove contents of attachments before writing to disk.
- Easily use with GMAIL/Yahoo/Hotmail servers requiring SSL/TLS.
- Start and terminate external programs from within an application.
- Can be used from GUI mode or console mode programs.
- Is fully thread safe.
- Implemented as a **standard** Windows DLL, which will work with all versions of Windows.
- Supports all versions of Alaska Xbase++, from V1.3 through V1.9.
- Can be used with Microsoft Visual .NET and Visual Studio.
- Does **not** depend on support libraries. Makes calls to core Windows API functions only.
- Can be used with any program (in any language) capable of calling Windows API functions such as Visual C++, Visual C++ NET, Visual FoxPro, Delphi, Xbase++, dBASE, COBOL, Access and Excel.
- Works with 32-bit and 64-bit Windows through Windows 10.
- License covers all programming languages.
- Royalty free distribution (no run-time fees) with your compiled application
- The evaluation version is fully functional.

Registration includes one year of free updates and technical support.

A good selection of XBase example programs with full source code is included. Refer to Section 6 for more details on each of the example programs.

[PROGRAM]		[DESCRIPTION]
SEEVER	:	Displays SEE Version/Build number and registration string.
AUTHEN	:	Uses authentication to connect to SMTP server.
AUTO	:	Auto-respond to email using 2 concurrent channels.
BCAST	:	Sends bulk email to one recipient per email.
CODETEST	:	Base64 encodes/decodes strings.
FORWARD	:	Forwards undecoded email.
FROM	:	Displays header information for email on server.
GB2312	:	Sends email that is GB2312 (simplified Chinese).
GETRAW	:	Downloads specified email without decoding.
HELLO	:	Emails a short message.
HTML	:	Sends html encoded email with attachments.
ISO8859	:	Sends ISO-8859 encoded message and subject line
MAILER	:	Sends email with optional attachment.
MAILER2	:	Sends email with optional attachment using the "indirect" method.
MailSSL	:	Connects to server requiring SSL to send email.
Mparts	:	Sends multipart MIME email.
POP3RD	:	Specifies email message file to read and decode.
READER	:	Downloads email & attachments and saves to disk.
READER2	:	Downloads email & attachments and saves to disk using the "indirect method."
ReadSSL	:	Downloads email from POP3 server requiring SSL.
STATUS	:	Lists all email on server and displays DATE, FROM and SUBJECT **header fields.**
STATUS2	:	Lists all email on server. Similar to STATUS but uses the "indirect method."
TESTCONN	:	Tests connection to a specified server and port.
ImapFlag	:	Tests manipulation of flaps on IMAP server.
ImapSrch	:	Tests IMAP search capability.

Also see EXAMPLES.TXT in the DOCS directory for a list of the examples provided for a particular compiler.

1.2 Documentation

The complete set of documentation consists of three manuals in Adobe PDF format. This is the first manual (SEE4XB) in the set.

- SEE4XB Programmer's Manual (SEE_4XB.PDF)
- SEE User's Manual (SEE_USR.PDF)
- SEE Reference Manual (SEE_REF.PDF)

The SEE_4XB Programmer's Manual (SEE_4XB) is the language specific (Xbase++) manual. All language dependent programming issues such as compiling, compilers and example programs are discussed in this manual. Read this manual first.

The SEE User's Manual (SEE_USR) discusses email processing as well as language independent programming issues. Purchasing and license details are also provided.

The SEE Reference Manual (SEE_REF) contains details on each individual **SEE** function as well as the **SEE library** error codes.

The online documentation can be accessed on the **SMTP/POP3/IMAP Email Engine for Xbase++** product page at:

http://www.marshallsoft.com/see4xb.htm

1.3 Example Program

The following example segment demonstrates the use of some of the **SMTP/POP3/IMAP Email for Xbase++** component library functions:

```
#INCLUDE "DLL.CH"
#INCLUDE "KEYCODE.CH"
#INCLUDE "SEE32.CH"

*** PROGRAMMER: Edit these strings [use host name or IP address for server] ***
SmtpServer = "10.0.0.1"
SmtpFrom = "<mike@10.0.0.1>"
SmtpReply = chr(0)
SmtpTo   = "<mike@10.0.0.1>"
LogFile = ".\HELLO.LOG"
*** END PROGRAMMER ***

? "HELLO 7.2"
Code = XseeAttach(1, SEE_KEY_CODE)
if Code < 0
  ? "Cannot attach SEE"
  return
endif
Code = XseeStringParam(0, SEE_LOG_FILE, @LogFile)
*** set maximum connect wait to 20 seconds
Code = XseeIntegerParam(0, SEE_CONNECT_WAIT, 20000)
*** connect to SMTP server
? "Connecting to " + SmtpServer
Code = XseeSmtpConnect(0, @SmtpServer, @SmtpFrom, @SmtpReply)
if Code < 0
  Temp = SPACE(128)
  Code = XseeErrorText(0,Code,@Temp,128)
  ? Left(Temp,Code)
else
  *** send email message
  ? "Sending email to " + SmtpTo
  Code = XseeSendEmail(0,SmtpTo,"","","Hello from Xbase++","Hello from Xbase++","")
  if Code < 0
    Temp = SPACE(128)
    Code = XseeErrorText(0,Code,@Temp,128)
    ? Left(Temp,Code)
  else
    ? "Email has been sent."
  endif
endif
? "Done."
Code = XseeClose(0)
Code = XseeRelease()
return
```

In the example program above, **seeAttach** is called to initialize **SEE** and then **seeSmtpConnect** is called to connect to the SMTP mail host. The SMTP server host name and your email address are required, while the "Reply-To" entry is optional.

seeSendEmail is then called, passing the addressee lists. The primary addressee is provided in the "To List". The CC ("Carbon Copy") lists additional recipients, as does the BCC (Blind Carbon Copy) list. The subject contains the email subject line. The message text is next. If it starts with the '@' symbol, it is considered the name of the file containing the email message. Lastly, the filename of any ASCII or binary attachment is specified. All fields, except the first, in **seeSendEmail** are optional.

After returning from **seeSendEmail**, the **seeClose** function is called to close the connection to the SMTP server. Lastly, **seeRelease** is called to perform **SEE** termination processing and release the Winsock.

1.4 Installation

(1) Before installation of SEE4XB, an Xbase++ compiler (any version) should already be installed on your system and tested.

(2) Unzip SEE4XB74.ZIP (evaluation version) or SEExxxxx.ZIP (registered version where xxxxx is your Customer ID) using any Windows unzip program.

(3) Run the installation program SETUP.EXE that will install all SEE4XB files and copy SEE32.DLL to the Windows directory.

The SETUP installation program creates four sub-directories (default \SEE4XB) as follows:

```
DOCS – All documentation files
APPS – All example code
DLLS – All DLL's
SSL  - Proxy server files (optional)
```

(4) You are ready to compile and run! For a quick start, load the project file SEEVER.PRG.

1.5 Uninstalling

Uninstalling SEE4XB is very easy.

First, run UINSTALL.BAT, which will delete SEE32.DLL from your Windows directory, typically C:\WINDOWS for Windows 95/98/Me/XP/2003/2012/Vista/Windows 7/Windows 8 or C:\WINNT for Windows NT/2000.

Second, delete the SEE4XB project directory created when installing SEE4XB.

1.6 Pricing

A developer license for the SMTP/POP3/IMAP Email Library can be registered for $115 USD. Purchasing details can be found in Section 1.4, "How to Purchase", of the SEE User's Manual (SEE_USR).

Also see INVOICE.TXT or

http://www.marshallsoft.com/order.htm

Registration includes one year of free updates and technical support. Registered DLLs never expire.

1.7 Updates

When a developer license is purchased, the developer will be sent a registered DLL plus a license file (SEExxxxx.LIC, where xxxxx is your Customer ID). The license file can be used to update the registered DLL for a period of one year from purchase. Updates can be downloaded from

http://www.marshallsoft.com/update.htm

After one year, your license must be updated to be able to download updates, although the registered DLL never expires. The license can be updated for $30 if ordered within one year of the original purchase (or previous update). Between one year and three years, licenses can be updated for $55. After three years, licenses can be updated for $75.

Note that registered DLL does not expire; however, the ability to download version updates expires after one year. Refer to the file UPDATES.TXT located in the /SEE4XB/DOCS directory for more information.

2 Library Overview

The **SMTP/POP3/IMAP Email** component library has been tested on multiple computers running Windows 95/98/Me/XP/2003/2008/2012/Vista/Windows 7/Windows 8 and Windows NT/2000.

2.1 Dynamic Link Libraries

The **SMTP/POP3/IMAP Email** component library includes a Win32 dynamic link library (DLL). A DLL is characterized by the fact that it need not be loaded until required by an application program and that only one copy of the DLL is necessary regardless of the number of application programs that use it. Contrast this to the traditional static library that is bound to each and every application that uses it at link time.

An important advantage that DLL's have over other "popular" library formats such as VBX or OCX is that DLL's are callable by all Windows applications. Since DLL's are the building blocks of the Windows Operating System, they will not be replaced by a "newer technology".

2.2 Keycode

The SEE32.DLL has a keycode encoded within it. Your keycode is a 9 or 10-digit decimal number (unless it is 0), and will be found in the file KEYCODE.CH. The keycode for the evaluation version is 0. You will receive a new keycode and a new SEE32.DLL after purchasing or updating a developer license. The KEYCODE must be passed to **seeAttach**.

If you get an error message (value -74) when calling **seeAttach**, it means that the keycode in your application does not match the keycode in the DLL. After registering and before installing the registered version, it is best to remove the evaluation version of the SEE32.DLL from the Windows search path or delete it.

2.3 Win32 STDCALL and DECLSPEC

SEE32 is written in ANSI C and is compiled using the _stdcall and _declspec keywords. This means that SEE4XB uses the same calling conventions and file naming conventions as the Win32 API. In particular, function names are NOT decorated. There are neither leading underscores nor trailing "@size" strings added to function names.

Any Windows application program may call the SEE32 library provided that the proper declaration file is used.

2.4 Error Display

The error message text associated with **SEE** error codes can be displayed by calling **seeErrorText**. Each sample program contains examples of error processing.

Also see the file seeErrors.txt for a list of all Winsock and SEE error codes.

2.5 Adding SEE4XB to a Project

It is straightforward to add **SEE** to both console mode and GUI Xbase++ programs. First, add

```
#INCLUDE "KEYCODE.CH"
#INCLUDE "SEE32.CH"
```

after any other `$INCLUDE` statements in the Xbase++ program.

Then add

```
nCode = seeAttach(1 SEE_KEY_CODE)
If nCode < 0 Then
  ? "Cannot attach SEE"
  return
endif
```

as the first executed **SEE** function.

The keycode (contained in KEYCODE.CH) is 0 for the evaluation version and is a 9-digit number for the purchased version. Rather than include KEYCODE.CH as shown above, the keycode can be pasted directly into the call to **seeAttach**.

Lastly, link your program with SEE32.LIB. Refer to the example programs in the SEE4XB/APPS subdirectory.

3 Compiler Issues

SEE4XB has been compiled and tested with Xbase++ version v1.3 through and Xbase++ v1.9. The SETUP installation program will copy the Lib's and SEE32.DLL to the Windows directory. Refer to Section 1.4 "Installation".

3.1 INCLUDE Files

All example programs include two files: `KEYCODE.CH` and `SEE32.CH`. The file `SEE32.CH` contains all the necessary constants and function declarations for SEE4XB, while the file `KEYCODE.CH` contains your keycode (license key), as discussed in Section 2.2.

The Alaska Xbase++ include file DLL.CH is also required. For example,

```
#INCLUDE "DLL.CH"
#INCLUDE "KEYCODE.CH"
#INCLUDE "SEE32.CH"
```

The above files can be copied to the Xbase++ INCLUDE directory (where Xbase++ can find it) if so desired.

Note that each function is declared with the prefix character of 'X'

3.2 Compiling and Linking Programs

Before compiling any of the example programs, edit each program with your TCP/IP email parameters, as shown in the example program in Section 1.3 above. Server names can be IP addresses (in decimal dot notation) or the host name. Email addresses must be enclosed in angle brackets.

More details on each of the example programs are provided in Section 4.0, "Example Programs.

To compile and link console mode programs, such as `STATUS.PRG`, use

```
XPP STATUS.PRG
ALINK STATUS.OBJ SEE32.LIB
```

To compile and link windows GUI programs, such as `FROM.PRG`, use

```
XPP FROM.PRG
ALINK /SUBSYSTEM:WINDOWS FROM.OBJ SEE32.LIB
```

3.3 Xbase++ Compiler

If you don't have the Alaska Software Xbase++ compiler, you can find it on the web at

http://www.alaska-software.com

4 Example Programs

Each example program, with the exception of SEEVER.PRG and CODETEST.PRG, must be edited with your TCP/IP email parameters before compiling. Refer to the SMTP/POP3/IMAP Email User's Manual (online at SEE_USR or /see4fp /DOCS/see_usr.pdf) for details regarding TCP/IP email parameters.

Refer to Section 3.2 above for information on compiling and linking the example programs.

Be sure to edit the example programs with your email parameters before compiling. Before writing your own programs, compile and run several of the example programs.

4.1 Connectionless Example Programs

The following **SEE4XB** example programs do not require a connection to a server.

4.1.1 SEEVER

This simple program displays the **SEE** library version number and registration string taken from SEE32.DLL. Its purpose is display the **SEE** version, build, and registration string as well as to verify that SEE32 is being found and loaded by Windows. The **SEEVER** program does not connect to your LAN (or the Internet).

This is the first program that you should compile and run.

4.1.2 CODETEST

The CODETEST example program demonstrates how to use **seeEncodeBuffer** and **seeDecodeBuffer**, which BASE64 encodes and decodes several test strings. The CODETEST example program also demonstrates the use of **seeEncodeUTF8** and **seeDecodeUTF8**.

4.1.3 TESTCONN

The TESTCONN example console mode program tests if a SMTP, POP3, or IMAP server is accepting connections on a specified port. This is very useful when attempting to connect to a new email server.

The user name and password are not used in order to connect to a server, but rather are used after the connection has been accepted by the server.

4.2 SMTP Email Example Programs

There are twelve SMTP email example programs. SMTP programs send email using an SMTP server.

4.2.1 AUTHEN

AUTHEN is an example program that connects to an SMTP server using SMTP Authentication. You must connect to a SMTP server that allows authentication.

AUTHEN.PRG must be edited with your email parameters before compiling.

4.2.2 AUTO

AUTO ("auto-responder") uses two channels concurrently to automatically respond to all new email. AUTO will read (but not delete) all email on your server and reply to each that "your email was received".

AUTO.PRG must be edited with your email parameters before compiling.

4.2.3 BCAST

The BCAST example program emails the same message (BCAST.TXT) to a list of addresses taken from the file, BCAST.EML, containing one email address per line. Along with your SMTP server and your email address, you must create the file containing the email message to send, and create another file containing the list of recipients.

4.2.4 Forward

The FORWARD example program forwards an email message to a new recipient. Only undecoded email messages can be forwarded.

Undecoded email message can be downloaded using the GETRAW and READER example programs.

4.2.5 GB2312

The GB2312 example program sends a text message that is GB2312 (simplified Chinese) encoded. The recipient's email client will be able to display the email message using the specified GB2312 character set provided that it is capable of identifying GB2312 MIME parts (such as MS Outlook).

4.2.6 HELLO

The HELLO program emails a short message. HELLO.PRG must be edited with your email parameters before compiling. Compare with the MAILER example program.

4.2.7 HTML

The HTML example program connects to an SMTP server and emails an HTML file (HTML.HTM) containing inline graphics (IMAGE1.GIF and IMAGE2.GIF). The graphics files are attached to the HTML email message.

HTML.PRG must be edited with your email parameters before compiling.

4.2.8 ISO8859

The ISO8859 example program sends a text message and subject line that are ISO-8859 encoded. The recipient's email client will be able to display the email message using the specified ISO character set provided that it is capable of identifying ISO-8859 MIME parts (such as MS OutLook).

4.2.9 MAILER

The MAILER example program emails a message, including an optional MIME attachment. MAILER.PRG must be edited with your email parameters before compiling.

4.2.101 MAILER2

The MAILER2 example program operates the same as the MAILER program, except that it uses the "indirect" method. MAILER2.PRG must be edited with your email parameters before compiling.

4.2.11 MailSSL

The MailSSL example program emails a specified email message connecting to a SMTP server that requires SSL, such as Gmail, Hotmail, and Yahoo. Be sure to read the section "Using Stunnel" in the SEE User's Manual (SEE_USR.PDF) in the SEE4XB\DOCS directory.

4.2.12 Mparts

The MParts example program sends a multipart MIME email in which the programmer specifies the Content-Type headers for each attachment.

The two attachment types specified in this example are a sound file (*.wav) and of PDF file (*.pdf).

4.3 POP3 Email Example Programs

There are nine POP3/IMAP example programs. These examples read email using a POP3 or IMAP server. Each example program, except for the FROM program, must be edited with your email parameters before compiling.

4.3.1 AUTO

AUTO ("auto-responder") uses two channels concurrently to automatically respond to all new email. AUTO will read (but not delete) all email on your server and reply to each that "your email was received".

AUTO.PRG must be edited with your email parameters before compiling.

4.3.2 FROM

The FROM program operates like the STATUS example program, except that it uses an Xbase++ form to enter the POP3 server, user, and password.

The FROM program must be linked as a GUI (rather than console mode) application:

```
XPP FROM.PRG
ALINK /SUBSYSTEM:WINDOWS FROM.OBJ SEE32.LIB
```

4.3.3 GETRAW

GETRAW is an example program that downloads a specified email message without decoding it (in "raw" format). This is used to see what the email looks like on the server. GETRAW.PRG must be edited with your email parameters before compiling.

4.3.4 Pop3Rd

The Pop3Read example program uses the seePop3Source function to specify an (undecoded) email message file to be read and decoded. There is no connection to any server.

Undecoded email message can be downloaded using the GETRAW and READER example programs.

4.3.5 READER

READER can read email, including multiple MIME attachments, from your POP3 server, deleting each email after being read. READER.PRG must be edited with your email parameters before compiling.

4.3.6 READER2

The READER2 example program operates the same as the READER program, except that it uses the "indirect" method. READER2.PRG must be edited with your email parameters before compiling.

4.3.7 ReadSSL

The ReadSSL example program downloads email messages from a POP3 server that requires SSL, such as Gmail, Hotmail, and Yahoo. Be sure to read the section "Using Stunnel" in the SEE User's Manual (SEE_USR.PDF) in the \SEE4XB\DOCS directory.

4.3.8 STATUS

STATUS reads the number of email messages waiting on your POP3 server, and displays the "DATE:", "FROM:", and "SUBJECT:" header fields from each email. STATUS.PRG must be edited with your email parameters before compiling.

4.3.9 STATUS2

The STATUS2 example program operates the same as the STATUS program, except that it uses the "indirect" method. STATUS2.PRG must be edited with your email parameters before compiling.

4.4 IMAP-Only Example Programs

There are two IMAP-only example programs. These examples access the IMAP server.

4.4.1 ImapFlag

The ImapFlag (IMAP Flags Test) example program tests the manipulation of flags on the IMAP server. It reads, sets, and deletes certain flags for the specified email message on the IMAP server.

IMAP flags are:

```
\Seen      Message has been read
\Answered  Message has been answered
\Flagged   Message is "flagged" for urgent/special attention
\Deleted   Message is "deleted" for removal by later EXPUNGE
\Draft     Message has not completed composition (marked as a draft).
\Recent    Message has arrived since the previous time this mailbox was
           selected. ["\Recent" may be fetched but not stored]
```

4.4.2 ImapSrch

The ImapSrch (IMAP Search) example program tests IMAP search capability.

See ImapSearch.txt or http://www.marshallsoft.com/ImapSearch.htm for a complete list of all IMAP search strings.

Example search strings as passed to seeImapSearch():

```
SEEN
SEEN NOT ANSWERED
FLAGGED SINCE 1-Feb-2008 NOT FROM "Smith"
LARGER 10000 NOT SEEN
```

5 Revision History

The **SMTP/POP3/IMAP Email Engine** DLL (SEE32.DLL) is written in ANSI C. All programming language versions of SEE (C/C++, Visual Basic, PowerBASIC, Delphi, Visual FoxPro, Visual dBase, Xbase++, COBOL, and FORTRAN) use the same identical DLL.

Version 3.2: February 21, 2000.

- Initial release of Xbase++ version.

Version 3.3: November 20, 2000
- seeGetEmailLines can use internal memory.
- Added SEE_COPY_BUFFER [seeDebug] to copy internal buffer.
- Added SEE_WRITE_TO_LOG [seeStringParam] to allow user to write to LOG file.
- Added SEE_GET_ATTACH_NAMES [seeDebug] to get attachment filename list.
- Ability to reset the SEE_SET_HEADER [seeStringParam] to "nothing".
- Added seeCommand function.
- Allow TIC marks (0x27) in VerifyAddressChars().
- Added SEE_GET_LAST_RECIPIENT to seeDebug.
- Added seconds to date string on outgoing email.
- Attachment name is saved when attachment file is closed.
- Added SEE_PATH_DELIMITER to seeIntegerParam().
- Added seeAbort function.
- VerifyFormat rejects "@domain" and "name@" addresses.
- Added "SEE_SET_FROM" so can change "From:" header at runtime.
- Delimiters (CR/LF) sent with command in one network transmission [seeWriteLine].
- Added QUOTED_USER, SEE_SET_CONTENT_TYPE, and SEE_SET_TRANSFER_ENCODING.
- Added SEE_ATTACH_DELIMITER and ability to specify different attachment filename in email.
- Added SEE_ADD_HEADER to seeStringParam.
- Added SEE_WRITE_BUFFER to seeDebug (see seeGetEmailLines)
- Added SEE_ENABLE_IMAGE to send GIF/TIF/BMP/JPG images inside email.

Version 3.4: August 14, 2001
- Supports "AUTH LOGIN" and "AUTH CRAM-MD5" (SMTP) authentication.
- SmtpResponse accepts response line without message.
- Supports ISO-8859-1 (base-64) encoding on subject line.
- Supports "APOP" authentication (POP3 servers).

Version 3.5: April 10, 2002

- Added support for "AUTH PLAIN".
- Recognize multiple AUTH methods on one line, such as "AUTH PLAIN LOGIN CRAM-MD5".
- Added SEE_FORCE_INLINE -- attachments are inline text rather than base64 encoded.
- Added SEE_SET_ATTACH_CONTENT_TYPE -- user can specify content type for attachments.
- Added SEE_ATTACH_BASE_NUMBER -- attachments named "1.att", "2.att", etc.
- Don't close socket (seeClose) if socket is already closed.
- NBR_CHANS set to 128 for Win32.
- SEE_RAW_MODE reads complete lines rather than buffers.
- Added seeQuoteBuffer() -- used to prepare ISO-8859 headers.
- Will continue with sending DATA (rather than return error) if have at least one recipient.
- Call seeStatistics(Chan, SEE_GET_LAST_RECIPIENT) to get # recipients accepted by server.
- Added SEE_IGNORE_REJECTED to ignore error returned if recipient is rejected.
- Added BCAST and CODETEST example programs.

Version 3.6: April 28, 2003

- Added seeSendHTML() function.
- Looks for multipart/related as well as multipart/alternative message parts.
- Added SEE_HTML_CHARSET (CHARSET_US and CHARSET_8859)
- Generic multipart boundary definitions handled (not just alternate, related, ...)
- CR/LFs preserved in multiline "Subjects:" headers.
- Handle case where "MIME-Version: 1.0" statement does not proceed all other MIME statements
- MAX_BUF increased from 2048 to 8192 for WIN32
- Virtual socket # written to log file when created (vsGetSocket) & released (vsCloseSocket).
- Write to email file if "MIME-Version" was not seen.
- vSock released properly in seeClose.
- Terminating ALT boundary not written if HTML file is passed from memory (not a file)
- Alternate text in seeSendHTML can be file (if prefixed with '@')
- Added seeEncodeUTF8 and seeDecodeUTF8 functions.
- Delimiters separating email addresses and pathnames changed to a semicolon.
- Added ISO_8859, WIN_1252, and WIN_1255 character set types.

Version 3.7: February 10, 2005.

- Terminating ALT boundary not written if HTML file is passed from memory (not a file).
- Alternate text in seeSendHTML can be file (if prefixed with '@')
- Added seeEncodeUTF8 and seeDecodeUTF8 functions
- AddrDelimiter and PathDelimiter changed to ';' (semicolon)
- Added QUOTED_WIN_1252 and QUOTED_WIN_1255.
- User headers written even if no subject
- Corrected problem: User Content-Type wasn't being sent if no quoting
- Added SEE_HIDE_HEADERS -- overrides any conflicting flags
- Fixed problem with "Filename=" extraction.
- Replaced OF_READ|OF_SHARE_DENY_WRITE with OF_SHARE_DENY_WRITE in _lopen
- Filename added to SEE_CANNOT_CREATE & SEE_CANNOT_OPEN error messages.
- Multi-line subject headers supported in seeGetEmailFile.
- ReadMsgLine uses Allow8Bits to decide if it should quote or not
- Added SEE_SET_DEFAULT_ZONE
- Increased buffer size for challenge string in authenticated SMTP connections.
- Added WriteToLog(), WriteClientTempToLog(), and WriteToLastLog() to centralize log writing.
- Nulls are replaced by spaces in all incoming data.
- Added support for "=?US-ASCII?B?" encoded filenames
- Fixed problem quoting line starting with '.' and having non-ASCII characters.
- Fixed SMTP problem when attaching large number of files (seeWriteSocket,seeWriteLine,seeWriteString).
- Added IgnoreErrorStatus (default TRUE) that skips socket error check in STATE_CONNECT
- Fixed problem with Content-Type prefix (set by SEE_WRITE_CONTENT_TYPE).
- Scan subjects & filenames for "big5" encoding like iso-8859
- Only one of TO, CC, and BCC must contain a recipient.
- Maximum text line length default increased to 1000.
- Added SEE_REPLACE_WITH_COMMAS to override replacement of delimiters with commas.
- SEE_FILE_PREFIX parameters set base for attachment file prefixes.
- Added seeAttachmentParams function.
- Added ISO8859, GB2312, and MParts example programs.

Version 4.0: July 24, 2006.

- Always an error if "relay", "gateway", or "not local" is in the text of the server's response, regardless of SEE_IGNORE_REJECTED.
- Forwarded header lines written to message/rfc822 (attachment) file.
- Each POP3 message optionally saved to disk in raw (undecoded) format in seeGetEmailFile.
- Added function seeForwardEmail().
- Added function seePop3Source().
- Maximum internal buffer size increased from 8 KB to 16 KB.
- Alternate boundaries w/o enclosing quotes are supported.
- FORWARD and Pop3Read example programs added.
- Added function seeByteToShort
- Added function seeShortToByte

Version 5.0: May 30, 2008 (Win32 Version only)

- Added seeSetErrorText.c example program
- Added LoadLib.c example program.
- Added IMAP capability. IMAP-only functions are:
  1. seeImapConnect : Connect to IMAP server.
  2. seeImapFlags : Get, set, or delete message flags.
  3. seeImapSearch : Search for messages with specified flags.
  4. seeImapMsgNumber : Gets message numbers from buffer filled by seeImapSearch.
  5. seeImapSelectMB : Selects IMAP mailbox.
  6. seeImapDeleteMB : Delete a mailbox.
  7. seeImapCreateMB : Create a new mailbox.
  8. seeImapRenameMB : Rename mailboxes.
  9. seeImapCopyMBmail : Copy messages from selected mailbox to specified mailbox.
  10. seeImapListMB : List all available mailboxes.
- Added ImapFlag and ImapSrch example programs.
- Pass NULL for filename to seePop3Source / seeImapSource to revert back to server processing.

Version 5.1: June 3, 2009

- Fixed code for IMAP_SEARCH_MSG_COUNT in seeImapMsgNumber
- Appended CR/LF to text returned by seeGetEmailUID
- Fixed problem with STATE_POP3_DELETE (call exiting via STATE_POP3_DELETE_OK)
- Added EnableHeaders to enable/disable writing of headers.
- Don't write blank line after headers (in STATE_SMTP_BODY) if EnableHeaders = 0
- Write the # bytes written to mail file in the log file.
- Never write boundaries to the email file.
- Fixed bug: seeGetEmailCount works with all IMAP mailboxes (not just InBox)
- Added seeStartProgram and seeKillProgram to start/terminate external programs.
- Fixed problem with blocking mode so connect timeout works.
- Added seeSmtpTarget that writes SMTP output to a file.
- Fixed problem with seeSendEmail (w/ attachment) after forwarding email.
- Added Win64 DLL to support x64. [Visual C++ and Visual Basic version].

Version 5.2: April 3, 2010

- Added seeSleep function (for languages not having a native Sleep call).
- The HELO command passes the computer name rather than its IP address.
- Bug Fix: All handles closed before memory blocks are freed.
- Bug Fix: Multiline "To:" header preserved in incoming email.
- Bug Fix: seeSmtpTarget now always closes files.
- Bug Fix: seePop3Source now always closes files.
- Bug Fix: Multiple IMAP response lines now handled properly by seeCommand.
- Added UTF8 character set support (CHARSET_UTF8).
- Added check for "MX lookup failure" when reading incoming mail.
- Added check for "Invalid MX record" when reading incoming mail.
- Changed IMAP list command argument default from ~/* to "" "*".
- Added SEE_SET_IMAP_LIST_ARG to seeStringParam (sets IMAP list command argument)
- Added seeReadQuoted function: reads a file and quotes the contents as it writes to a buffer.
- Added "Buffer overflow" error code.
- Added QUOTED_ISO_8859_2 to seeIntegerParam for sending ISO_8859_2 encoded emails.
- Added QUOTED_ISO_8859_7 to seeIntegerParam for sending ISO_8859_7 encoded emails.
- Added SEE_GUT_ATTACHMENTS to seeIntegerParam to remove contents of incoming attachments.

Version 6.0: February 21, 2011

- Better integration to the Stunnel proxy server.
- Added seeSmtpConnectSSL and seePop3ConnectSSL.
- Added seeIsConnected.
- Fixed: Can now have leading period in alternate text.
- Added SEE_SET_LOCAL_IP (seeStringParam) to specify local IP.
- Added CHARSET_WIN_1250.
- Changed (default) MaxResponseWait from 10 secs to 25 secs.
- Added SEE_SET_HELO_STRING.
- Fixed problem with reading POP3 from file.
- Add support for ISO-8859-3 and ISO-8859-4.

Version 7.0: November 10, 2011

- Fixed problem decoding some "ISO-8859" subjects
- Fixed problem with wrong content type when using seePop3Rd
- Fixed problem with seeAttachmentParams
- Added seeImapConnectSSL()
- ParseISO removes iso-8859-15 encoding from incoming Subject, etc.
- "To:" and "CC:" strings decoded (base64 & quoted)
- Decode quoted UTF-8 subject strings
- Replace underscore with blank (RFC2047) in UnQuote
- Added ".png" to image types
- Call seeStringParam(Chan, SEE_SET_HELO_STRING, '*') to use machine name for HELO string
- Call seeStringParam(Chan, SEE_LOG_FILE, "\0") to disable logging
- Recognizes iso-2022-jp
- Added seeSetProxySSL()
- Modified seeSmtpConnectSSL(), seePop3ConnectSSL(), seeImapConnectSSL(). Includes changes so that Stunnel (used for email services requiring SSL) is automatically configured, loaded, and unloaded without any user intervention.
- Use large buffer (64K) for IMAP server response on channel 0.

Version 7.1: April 12, 2012

- Can pass full pathname for ProxyEXE and ProxyCert in seeSetProxySSL.
- Buffer sizes for ProxyEXE & ProxyCert (seeSetProxySSL) increased from 64 to 256 chars.
- (NOTE: can an no longer pass a null string for PEM certificate)
- seeRelease() kills all running copies of Stunnel started by SEE.
- Password characters not written to log file (PASS ****) & AUTH transmissions
- Added SEE_SET_CONNECT_ATTEMPTS that sets max connection attempts (1 to 12)
- Fixed problem: ImapConnect not returning error if bad login.
- SEE closes all process handles for all external program started by SEE.

Version 7.2: September 20, 2013

- Increased the maximum number of channels from 32 to 64.
- Allow multiple subject lines in incoming email.
- Added SEE_REPLACE_UNDERSCORES to seeIntegerParam() to disable replacement of underscores with spaces (RFC2047).
- Fixed problem with GMAIL IMAP connection.
- Can now decode Win1255 subjects.
- seeAbort now always closes attachment files.
- Fixed zone calculation for "half-zones".
- Added debug info to seeGetEmailCount().
- Added STUNNEL_DISABLE_LOGGING flag to seeSetProxySSL() that disables Stunnel logging.
- Fixed problem with SEE_ADD_HEADER when re-opening connection.
- Allow attachment filename to have a leading space.
- Added seeGetHeader() function with parameters SEE_GET_SUBJECT, SEE_GET_FROM, SEE_GET_REPLT_TO, SEE_GET_TO, and SEE_GET_DATE

Version 7.3: December 16, 2014

- Decodes UTF8 encoded attachment filenames.
- Diagnostics written to log file if missing '<' or '>' delimiters in email addresses.
- Added SEE_ALLOW_PARTIAL to seeIntegerParam which allows PARTIAL commands in IMAP.
- Added SEE_GET_UIDVALIDITY to seeStatistics which returns UID Validity in IMAP.
- Fixed problem with boundary buffer [64-bit only].
- Added seeConfigSSL() function which adds lines to the SSL configuration file.
- Added seeUnquote() function that unquotes quoted buffers.
- Added UTF8 quoting : seeIntegerParam(Chan, SEE_QUOTED_PRINTABLE, QUOTED_UTF8)

Version 7.4: April 6, 2016

- Changed: seeImapConnect() & seeImapConnect() now hide LOGIN password .
- Added: Content-Type marked automatically for PDF and WAV files.
- Fixed: socket forced closed if cannot connect to server.
- Fixed: replace non ASCII characters in the subject and header strings with the '_' character.
- Added: allow commas to be used in a filename itself (seeTestFileSet).
- Added: seeMakeSubject() to make ISO & UTF-8 quoted subject strings.
- Added: more diagnostics to the SEE log file.
- Added: new example program TestConn.prg that tests connection to server.