SMTP/POP3/IMAP Email Engine
Library for PowerBASIC

Programmer's Manual

(SEE4PB)

Version 8.5

November 7, 2024

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 PowerBASIC (SEE4PB)** library is a toolkit that allows software developers to quickly develop SMTP and POP3/IMAP email applications in PowerBASIC (PB/CC, PB/DLL, and PB/WIN).

A straightforward interface allows sending, receiving and parsing 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 Email Engine (SEE)** is a component library of functions providing direct and simple control of the SMTP (Simple Mail Transport Protocol), POP3 (Post Office 3), and IMAP 4 (Internet Message Access Protocol) protocols.

The **SMTP/POP3/IMAP Programmer's Manual** provides information needed to compile and run programs in a PowerBASIC programming environment.

The **SMTP/POP3/IMAP Email Engine for PowerBASIC** component library supports and has been tested with the PowerBASIC console compiler (PB/CC) and PowerBASIC for Windows (PB/DLL and PB/WIN). **SEE4PB** includes numerous example programs that demonstrate SMTP and POP3/IMAP used to build mail applications using the SEE4PB library.

SEE4PB 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, Xbase++, dBASE, etc.) capable of calling the Windows API.

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

- SEE4PB is a standard Windows DLL and is much smaller than an OCX or ActiveX control.
- SEE4PB does NOT depend on ActiveX or similar "support" libraries.
- SEE4PB (SEE32.DLL) is fully threadable.
- The SEE4PB functions can be called from applications not capable of using controls.

MarshallSoft also has versions of the **SMTP/POP3/IMAP Email Engine** library for Visual Basic (SEE4VB), Delphi (SEE4D), C/C++ (SEE4C), Visual FoxPro (SEE4FP), COBOL (SEE4CB), Visual dBASE (SEE4DB), and Xbase++ (SEE4XB). All versions of the **SEE** library use the same DLLs (SEE32.DLL or SEE64.DLL). However, the examples provided for each version are written for the specified programming language.

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/IMAP client 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 server.
- Receive any email on the POP3/IMAP server including MIME attachments.
- Remove contents of attachments before writing to disk.
- 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/IMAP 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.
- Supports CHARSET_WIN_1250, CHARSET_WIN_1252 and CHARSET_WIN_1255.
- Can specify custom Content-Types; add custom header fields
- Includes 70 functions for SMTP and POP3/IMAP mail control.
- Dozens of switches provided to control how email is sent or received.
- Supports setting priority via X-Priority header field.
- Use with GMAIL /Yahoo/Hotmail servers requiring SSL with (free) STUNNEL proxy server.
- 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 PowerBASIC Console Compiler (PB/CC), PowerBASIC for Windows (PB/WIN) and PB/DLL.
- 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 C++, Visual C++ NET, dBASE, Delphi, Xbase++, Visual Basic, COBOL, Access and Excel..
- Works with Windows XP through 32-bit and 64-bit Windows 11.
- License covers all programming languages
- The evaluation version is fully functional.
- License includes technical support and free updates for one year.
- Royalty free distribution (no run-time fees) with your compiled application.

A good selection of PowerBASIC 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. (PBCC)
 AUTHEN    :  Uses authentication to connect to SMTP server. (PBCC)
 AUTO      :  Auto-respond to email using 2 concurrent channels. (PBCC)
 BCAST     :  Sends bulk email to one recipient per email. (PBCC)
 CODETEST  :  Base64 encodes/decodes strings. (PBCC)
 DELETE    :  Deletes an email message from server. (PBCC)
 FORWARD   :  Forwards undecoded email. (PBCC)
 FROM      :  Displays header information for email on server. (PBCC)
 GB2312    :  Sends email that is GB2312 (simplified Chinese). (PBCC)
 GETRAW    :  Downloads specified email without decoding. (PBCC)
 wHello    :  Reports the SEE version.  (PBWIN)
 HTML      :  Sends html encoded email with attachments. (PBCC)
 ISO8859   :  Sends ISO-8859 encoded message and subject line.  (PBCC)
 MAILER    :  Sends email with optional attachment. (PBCC)
 wMailer   :  Sends email with optional attachment. (PBWIN)
 MailSSL   :  Connects to server requiring SSL to send email.  (PBCC)
 wMailSSL  :  Connects to server requiring SSL to send email.  (PBWIN)
 Mparts    :  Sends multipart MIME email.  (PBCC)
 POP3RD    :  Specifies email message file to read and decode.  (PBCC)
 PB_STAT   :  Calls DLL created by PBDLL from Microsoft Visual Basic. (PBDLL)
 READER    :  Downloads email & attachments and saves to disk. (PBCC)
 ReadSSL   :  Downloads email from POP3 server requiring SSL. (PBCC)
 wReadSSL  :  Downloads email from POP3 server requiring SSL. (PBWIN)
 wReader   :  Downloads email from POP3 server. (PBWIN)
 STATUS    :  Lists all email on server. Displays DATE, FROM, SUBJECT. (PBCC)
 TSTAT     :  Checks several POP3 mailboxes using threads. (PBCC)
 TestConn  :  Tests connection to a specified server and port (PBCC).
 VERUSR    :  Requests verification of user on SMTP server. (PBCC)
 ImapFlags :  Tests manipulation of flaps on IMAP server. (PBCC)
 ImapSearch:  Tests IMAP search capability. (PBCC)
```

1.2 Documentation Set

The complete set of documentation consists of three manuals in Adobe PDF format. This is the first manual (SEE_4PB) in the set.

- SEE4PB Programmer's Manual `(SEE_4PB.PDF)`
- SEE User's Manual `(SEE_USR.PDF)`
- SEE Reference Manual `(SEE_REF.PDF)`

The SEE_4PB Programmer's Manual is the language specific (PowerBASIC) manual. All PowerBASIC 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.

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

    http://www.marshallsoft.com/see4pb.htm

1.3 Example Program

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

```
Dim Code As Integer
Dim xServer, xIsNull, xTo, xFrom As Asciiz * 64
Dim xSubject, xMessage As Asciiz * 64
xIsNull  = Chr$(0)
xServer  = "mail.yourisp.com
xFrom = "my name<me@myisp.biz>"
xTo    = "<support@marshallsoft.com>"
xSubject = "PowerBASIC Test"
xMessage = "Emailed from SEE4PB!"
Code = seeAttach(1, %SEE_KEY_CODE)
If Code < 0 Then
  ' error calling seeAttach !
  . . .
End If
Code = seeSmtpConnect(0,xServer,xFrom,xIsNull)
If Code < 0 Then
  ' error calling seeSmtpConnect !
  . . .
End If
Code = seeSendEmail(0,xTo,xIsNull,xIsNull,xSubject,xMessage,xIsNull)
If Code < 0 Then
  ' error calling seeSendEmail !
  . . .
End If
Code = seeClose(0)
Code = seeRelease()
```

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 SEE4PB, a PowerBASIC compiler (PB/CC or PB/WIN) should already be installed on your system and tested.

(2) Unzip SEE4PB851.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 which will install all SEE4PB files, including copying SEE32.DLL to your Windows directory.

(4) Edit the INCLUDE file EMAIL.INC (which is included by most of the example programs) with your TCP/IP parameters.

(5) You're ready to compile the example programs. Compile and run SEEVER.BAS first.

1.5 Uninstalling

Uninstalling SEE4PB is very easy.

First, run UINSTALL.BAT, which will delete SEE32.DLL from your Windows directory, normally C:\WINDOWS

Second, delete the **SEE** project directory created when installing SEE4PB.

1.6 Pricing

A developer license for the SMTP/POP3/IMAP Email Library can be registered for $139 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's for a period of <u>one year</u> from purchase.   Updates can be downloaded from

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

After one year, the developer license must be updated to be able to download updates and receive technical support. The license can be updated for:
- $33 if the update is ordered within one year of the original purchase (or previous update).
- $55 is the update is ordered between 1 and 3 years of the original purchase (or previous update).
- $77 if the update is ordered after three years of the original purchase (or previous update).

The update price includes **technical support** and downloadable updates for an additional year. Note that the registered DLLs, (SEE32. DLL and SEE64.DLL) never expire. Refer to the file UPDATES.TXT located in the /SEE4PB/DOCS directory for more information.

2 Library Overview

The **SMTP/POP3/IMAP Email** component library has been tested on multiple computers running Windows XP though Windows 11.

The SEE4PB library has been tested and works with the PowerBASIC Console Compiler (PBCC), the PowerBASIC DLL Compiler (PBDLL) and PowerBASIC for Windows (PB/WIN).

The SETUP installation program will copy the SEE32.DLL to the Windows directory and copies the SEE4PB files to the directory specified (default \SEE4PB). Three sub-directories are created, as follows:

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

2.1 Dynamic Link Libraries

The **SEE4PB SMTP/POP3/IMAP Email** library component (SEE32.DLL) is implemented as a Win32 dynamic link libraries (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 DLLs have over other "popular" library formats such as VBX or OCX is that DLLs are callable by all Windows applications. Since DLLs are the building blocks of the Windows Operating System, they will not be replaced by a "newer technology".

2.2 Keycode

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.PBI. The keycode for the evaluation (shareware) version is 0. You will receive a new keycode and a new SEE32.DLL after purchasing or updating a developer license. The KEYCODE is 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, 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 SEE4PB uses the same calling conventions and file naming conventions as the Win32 API. . In particular, function names are NOT decorated; Leading underscores and trailing "@size" strings are not added to function names.

The SEE32.DLL functions may be called by any Windows application program capable of calling the Windows API provided the proper declaration file is used.

2.4 Using Threads

SEE4PB is thread safe, and can be used from any Windows application capable of using threads. Refer to the TSTAT example program, which demonstrates the use of threads.

The necessary Win32 thread functions are declared in the PowerBASIC INCLUDE file "WIN32API.INC"

2.5 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.

2.6 Editing EMAIL.INC

The file EMAIL.INC is included by most of the example programs and contains the following definitions:

```
#define YOUR_EMAIL_ADDR      "!your_email_address
#define SMTP_HOST_NAME       "!smtp_server_name"
#define SMTP_USER_NAME       "!smtp_account_name"
#define SMTP_PASSWORD        "!smtp_account_password"
#define SMTP_PROXY_PORT      0
#define POP3_HOST_NAME       "!pop3_server_name"
#define POP3_USER_NAME       "!pop3_account_name"
#define POP3_PASSWORD        "!pop3_account_password"
#define POP3_PROXY_PORT      0
```

This file must be edited with your email configuration before compiling any of the example programs. This information is required for all email clients. If you are unsure of any of these settings, look in your email client program (Outlook, Eudora, Pegasus, etc.).

SMTP_USER_NAME and SMTP_PASSWORD are required **only** if you will be doing SMTP authentication. They are usually the same as POP3_USER_NAME and POP3_PASSWORD. Refer to the SEE User's Manual (SEE_USR) for more information on authentication.

SMTP_HOST_NAME is the name (or IP address) of your SMTP email server. POP3_HOST_NAME is the name of your POP3 server, which is typically the same as the SMTP server. YOUR_EMAIL_ADDR is your email address. Brackets "<>" are required. POP3_USER_NAME is the account name for your email (POP3) account. POP3_PASSWORD is the password for you email (POP3) account. For example,

```
#define YOUR_EMAIL_ADDR    "Mike Marshall<info@marshallsoft.com>"
#define SMTP_HOST_NAME     "mail.yourisp.net"
#define SMTP_USER_NAME     "marshallsoft"
#define SMTP_PASSWORD      "qwerty"
#define POP3_HOST_NAME     "mail.hiwaay.net"
#define POP3_USER_NAME     "marshallsoft"
#define POP3_PASSWORD      "qwerty"
#define SMTP_USER_NAME     "marshallsoft"
#define SMTP_PASSWORD      "qwerty"
```

SMTP_PROXY_PORT and POP3_PROXY_PORT are used to communicate with the Stunnel proxy server, used to provide SSL services to Windows when connecting to email servers such as Gmail, Hotmail, and Yahoo.

Read the section "Using Stunnel" in the SEE User's Manual (SEE_USR.PDF) in the DOCS directory or online at http://www.marshallsoft.com/stunnel.htm

Note that EMAIL.INC is used only in the example programs. Strings passed to SEE functions can be taken from any source (hard-coded, read from keyboard, read from file, etc.).

Not all of the #defines are used by the example programs. For example, SMTP_USER_NAME and SMTP_PASSWORD are used only by the AUTHEN example program.

2.7 Adding SEE to a PowerBASIC Program

In order to add **SEE** to a PowerBASIC program, add

```
$INCLUDE "SEE32.PBI"
$INCLUDE "KEYCODE.PBI"
```

after any other $INCLUDE statements. Then add

```
Code = SeeAttach(1,%SEE_KEY_CODE)
If Code < 0 Then
  '... display error code to user here...
  EXIT FUNCTION
END IF
```

as the fist executed SEE function.

2.8 PowerBASIC Declaration Files

SEE constants are defined and all **SEE** functions are declared in the file SEE32.PBI and are included in all of the example programs.

2.9 Include Files

There are three include files. The first two are included by all example programs.

- SEE32.PBI -- Contains the function declarations for SEE32.DLL.
- KEYCODE.PBI -- Contains your **key code**. See the following section.
- GETARGS.PBI -- Read and parses command line arguments.

3 Compiler Issues

The **SMTP/POP3/IMAP Email Engine for PowerBASIC** component library supports and has been tested with the PowerBASIC Console Compiler (PBCC), the PowerBASIC DLL Compiler (PBDLL), and PowerBASIC for Windows Compiler (PBWIN).

3.1 PowerBASIC Console Compiler (PBCC)

Most of the example programs are written for the PowerBASIC Console Compiler (PBCC). Be sure to read the comments at the beginning of each example program. Also see Section 4.0 "Example Programs" below.

3.2 PowerBASIC DLL Compiler (PBDLL)

SEE functions can be called from programs written for the PowerBASIC DLL Compiler (PBDLL). The same DLL (SEE32.DLL) is used for both PowerBASIC and Visual Basic. **SEE** functions are defined in SEE32.PBI for PowerBASIC and in SEE32.BAS for Visual Basic. The Visual Basic version can be downloaded from http://www.marshallsoft.com/see4vb.htm

See the PB_STAT.BAS example program.

3.3 PowerBASIC Windows Compiler (PBWIN)

SEE functions can be called from programs written for the PowerBASIC Windows Compiler (PBWIN). There are no differences in how **SEE** functions are called in programs written in any of the Windows PowerBASIC compilers.

See the wHello.bas, wMailer.bas, and wReader.bas, wMailSSL.bas, and wReadSSL.bas example programs.

3.4 Compiling Example Programs

There are 30 example programs for PBCC, PBDLL, and PBWIN.

3.4.1 PowerBASIC Console Compiler (PBCC)

PBCC program are compiled from the command line. For example:

```
PBCC   AUTHEN.BAS
PBCC   AUTO.BAS
PBCC   BCAST.BAS
PBCC   CODETEST.BAS
PBCC   DELETE.BAS
PBCC   FORWARD.BAS
PBCC   GB2312.BAS
PBCC   GETRAW.BAS
PBCC   MailSSL.BAS
PBCC   ReadSSL.BAS
PBCC   HTML.BAS
PBCC   IMAPFLAGS.BAS
PBCC   IMAPSEARCH.BAS
PBCC   ISO8859.BAS
PBCC   MAILER.BAS
PBCC   MPARTS.BAS
PBCC   POP3RD.BAS
PBCC   READER.BAS
PBCC   SEEVER.BAS
PBCC   STAT.BAS
PBCC   TestConn.BAS
PBCC   STATUS.BAS
PBCC   TSTAT.BAS
```

3.4.2 PowerBASIC DLL Compiler (PBDLL)

PBDLL programs are compiled from the command line. For example:

```
PBDLL PB_STAT.BAS
```

3.4.3 PowerBASIC Windows Compiler (PBWIN)

PBWIN program are compiled from the PBWIN environment. To load your program, from the menu bar click "File", then "Open File". To compile , click "Run", then "Compile and Execute".

The five PBWIN example programs are

```
wHello.bas   -- Reports SEE version.
wMailer.bas  -- Sends email message.
wReader.bas  -- Reads email message.
wMailSSL.bas -- Sends email message (after SSL connection).
wReadSSL.bas -- Reads email message (after SSL connection).
```

4 Example Programs

Before compiling the example programs, edit the include file EMAIL.INC with your SMTP/POP3/IMAP host and email address. Refer to Refer to Section 2.6, Editing Email.Inc, for more information on SMTP/POP3/IMAP server names.

All example programs are compiled with PBCC except for PB_STAT.BAS that is compiled with PBDLL and wHello, wMailer, wReader, wMailSSL, and wReadSL which are compiled with PBWIN.

4.1 Connectionless Example Programs

Three example programs do not require a connection to a server.

4.1.1 SEEVER

The SEEVER example program displays the SEE library version number and registration string and verifies that SEE32.DLL is being found and loaded by Windows. Compile with PBCC.

4.1.2 CODETEST

The CODETEST example program demonstrates how to use **seeEncodeBuffer** and **seeDecodeBuffer**, which BASE64 encodes and decodes a test string. The CODETEST example program also demonstrates the use of **seeEncodeUTF8** and **seeDecodeUTF8**. Compile with PBCC.

4.1.3 wHELLO

The wHello is the same program as SEEVER except that it is compiled with the PowerBASIC Windows (PBWIN) compiler.

4.1.4 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 13 SMTP email example programs. SMTP programs send email using an SMTP server. Don't forget to edit EMAIL.INC before compiling.

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 takes no command line arguments.

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 (and TSTAT) are the only example programs that use more than one channel concurrently.

4.2.3 BCAST

BCAST (Broadcast) emails the same message to each recipient from a file of email addresses. 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. See BCAST.EML for an example.

4.2.4 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).

4.2.5 HTML

The HTML example program connects to an SMTP server and emails an HTML file containing inline graphics. The graphics files are attached to the HTML email message.

To email the test file (TEST.HTM with image attachments IMAGE1.GIF and IMAGE2.GIF) to address <you@my-isp.com>, type

```
HTML  you@my-isp.com
```

Note that the HTML example program uses the **seeSendHTML** function rather than the **seeSendEmail** function to send the email.

4.2.6 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).

4.2.7 MAILER

MAILER emails a message, including optional MIME attachments. All required parameters are input from the command line at runtime.

For example, to mail TEST.TXT with attachment TEST.ZIP to info@marshallsoft.com with subject "TEST" (Note that <> brackets are required around the email address.):

```
MAILER  "@test.txt"  TEST  "<info@marshallsoft.com>" test.zip
```

4.2.8 MPARTS

The MPARTS example program sends a multipart MIME email in which the Content-Type headers for each attachment are specified by the programmer.

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

4.2.9 VERUSR

VERUSR (Verify User) connects to a specified SMTP server and requests verification of the user.

Due to security concerns, some SMTP servers will not honor a "verify user" request. A user that does not verify does NOT necessarily mean that the email address is not good.

4.2.10 Forward

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

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 DOCS directory or online at http://www.marshallsoft.com/stunnel.htm

4.2.12 wMailer

The wMailer example program is a PBWIN program that sends an email. All email parameters are entered into text boxes are runtime.

4.2.13 wMailSSL

The wMailSSL example program is a PBWIN program that sends an email after connecting to a SMTP server that requires SSL. All email parameters are entered into text boxes are runtime.

4.3 POP3/IMAP Example Programs

There are 12 POP3/IMAP email example programs. These examples read email from a POP3 (or IMAP) server. Don't forget to edit EMAIL.INC 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 (and TSTAT) are the only example programs that use more than one channel concurrently.

4.3.2 DELETE

The DELETE program deletes an email message from the server. Be careful! Once deleted, email messages cannot be recovered.

4.3.3 GETRAW

GETRAW is a program that downloads a specified email message without decoding it. This is used to see what the email looks like on the server. Email can also be downloaded without decoding by the READER example program.

For example, to download message 2 from the POP3 account mail.myisp.biz as user "mike" and password "qwerty", type

```
GETRAW  "mail.myisp.biz"  "mike"  "qwerty"  2
```

4.3.4 PB_STAT

The PB_STAT example program demonstrates how to call DLL's created by the PowerBASIC DLL compiler from Microsoft Visual Basic. PB_STAT.BAS is compiled by

```
PBDLL PB_STAT.BAS
```

You must also compile the Visual Basic program PB_STAT.FRM by loading PB_STAT.VBP in Visual Basic version 4.0 or above.

4.3.5 READER

READER reads an email message from your server, including any MIME attachments. The email message is saved as a file.

For example, to read message 1 to file FILE1.TXT, type

```
READER  1  FILE1.TXT
```

4.3.6 STAT

The STAT program is the simplest POP3 example. It logs onto the POP3 server and gets the number of messages waiting.

STAT takes no command line arguments.

4.3.7 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 takes no command line arguments.

4.3.8 TSTAT

TSTAT checks several POP3 mailboxes using multiple threads. Edit TSTAT.BAS with the POP3 accounts to check before compiling. Note that since TSTAT references the WINAPI directory, it must appear on the command line. For example (assuming PBCC is installed at C:\PBCC):

```
PBCC TSTAT.BAS -IC:\PBCC\WINAPI
```

TSTAT takes no command line arguments. Be sure to edit TSTAT.BAS before compiling.

4.3.9 POP3RD

The POP3RD example program uses the seePop3Source function to specify an (undecoded) email message file to be decoded.

4.3.10 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 DOCS directory or online at http://www.marshallsoft.com/stunnel.htm

4.3.11 wReader

WReader is a PBWIN program that downloads an email. All email parameters are entered into text boxes are runtime.

4.3.12 wReadSSL

wReadSSL is a PBWIN program that downloads an email after connecting to a POP3 server that requires SSL. All email parameters are entered into text boxes are runtime.

4.4 IMAP-Only Example Programs

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

4.4.1 ImapFlags

The ImapFlags 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 ImapSearch

The ImapSrch 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 language versions of SEE (C/C++, Delphi, Visual Basic, PowerBASIC, FoxPro, Delphi, Xbase++, and COBOL) use the same identical DLLs.

Version 1.0: June 22, 1998.

- Initial release (for C/C++).

Version 2.0: October 2, 1998.

- Initial PBCC release.

Version 2.1: December 9, 1998.

- Time zone calculated automatically.
- Fixed bug in seeClose.
- Corrected POP3 problem when boundary definition on 2nd line.
- Added support for alternate MIME boundaries.
- Added seeVerifyUser function.
- Added SEE_GET_REGISTRATION and SEE_GET_CONNECT_STATUS.
- Added SEE_GET_ATTACH_COUNT and SEE_GET_LAST_RESPONSE.
- Added GETDOC, SEEVER, and VERUSR example programs.
- SMTP performance improved.
- Added seeEncodeBuffer and seeDecodeBuffer functions.
- Added SEE_FILE_PREFIX and SEE_SET_REPLY parameters.

Version 3.0: April 5, 1999.

- Modified SEE to be fully threadable (adding seeAttach and seeRelease).
- Added seeGetEmailUID function.
- Handles "inline" email text properly.
- Optionally decodes unnamed attachments.
- Added ability to add header lines (SEE_SET_HEADER).
- Can use alternate ports for SMTP or POP3.
- Win16 version can get time zone from TZ environment variable.
- Quoted-printable messages can handle soft line breaks.
- Quoted-printable message can handle embedded HTML text.

Version 3.1: July 21, 1999.

- Support ISO-8859-1 (Q or B) encoded attachment filenames.
- Support ISO-8859-1 (Q only) on subject line
- SEE_SAVED_TO_MSG parameter added.
- "+OK" line not written to email message file.
- Added seeExtractLine function.
- Added RegMe example programs.

- Added QUOTED_8859 processing (QUOTED_8859).
- Can decode printed quotable attachments!
- seeGetEmailLines can use internal memory.
- Added SEE_WRITE_TO_LOG to seeStringParam.
- Added SEE_GET_ATTACH_NAMES to seeDebug to get attachment filename list.
- Ability to reset the SEE_SET_HEADER header string to "nothing".
- Improvements in dynamic memory usage.
- Added GETRAW and CODETEST examples.
- Added support for Lcc-Win32.
- Added seeCommand function.

Version 3.3: October 30, 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, 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: March 20, 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 PB_STAT and CODETEST example programs.

Version 3.6: April 1, 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 4, 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: June 27, 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 Pop3Rd example programs added.
- Added function seeByteToShort
- Added function seeShortToByte

- 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 10, 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 26, 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: March 17, 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.
- Added WReader.BAS example program (PBWIN).
- Added MailSSL.BAS and ReadSSL.BAS example programs.

Version 7.0: November 30, 2011

- Added wMailSSL and wReadSSL example programs (PBWIN).
- 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 17, 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 5, 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 4, 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 21, 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.vb that tests connection to server.

Version 7.5: October 11, 2017

- Fixed : Problem fixed in which two user headers can't be set.
- Added : SEE_SET_RCPT_TRACE_FILE added to seeStringParam to write RCPT trace to disk
- Added : Added seeSetCertAuth() to specify Certification Authority certs (for SSL)
- Added : Current directory filename is written to log file
- Added : The date stamp filename is written to the log file
- Added : The SEE version written to log file moved to just after log file is opened
- Added: Writes SSL file date stamps to SEE log file.

Version 8.0: November 8, 2018

- Fixed: Multiple subject lines were not always correctly combined (seeGetEmailFile).
- Fixed: Problem with SSL PLAIN authentication protocol corrected.
- Change: Base64 strings passed to seeDecodeBuffer() no longer required to end with CRLF.
- Change: Increased value of MaxConnectAttempts from 12 to 20.
- Added: Constant SEE_SHOW_PASSWORDS added that shows all passwords in log file.
- Added: Added function seeEncodeUTF8String() that encodes UTF8 strings.
- Added: Added function seeDecodeUTF8String() that decodes UTF8 strings.
- Added: Added constant SEE_GET_CUSTOMER_ID to function seeStatistics().
- Added: Added error code constant SEE_BAD_UTF8_CHAR.
- Added: Customer ID written to Stunnel configuration file.
- Added: Additional connection statistics written to SEE log file.
- Added: SEE writes SSL file date stamps to SEE log file.

Version 8.1: April 8, 2020

- Fixed: Fixed seeTestFileSet() problem (file critical section not initialized)
- Added: Added Windows error text written to log file when Windows returns an error
- Change: Files opened for read now in shared access (FILE_SHARE_READ)
- Fixed: Fixed "CID=" string in log file
- Added: IMAP response "+OK" accepted in addition to "OK"
- Fixed: Fixed problem with UnQuote (did not handle lower case hex correctly)
- Added: Added error code SEE_INDEX_RANGE
- Added: Added function seeSetFileBuffer() - used to pass buffer (not file) as attachment
- Added: Added BASE64_UTF8 to seeMakeSubject()
- Added: seeStartProgram() sets last error for subsequent call to seeDebug(Chan, SEE_GET_LAST_ERROR,...)
- Added: Added SEE_GET_LAST_ERROR (36) to seeDebug
- Added: Error text written to log file if Stunnel can't be started
- Fixed: Handles decoding ISO & UTF8 multi-line subjects correctly
- Added: Writes KeyCode value to log file if seeDebug(...,SEE_GET_REGISTRATION,...) is called
- Change: Change SEE_REPLACE_UNDERSCORES (EnableRFC2047) default to FALSE
- Added: Added SEE_GET_KEYCODE to seeStatistics()
- Change: Multi-line "Subject:" lines saved as lines (not concatenated) for retrieval by seeGetHeader()
- Change: Multi-line "To:" lines saved as lines (not concatenated) for retrieval by seeGetHeader()

Version 8.2: January 21, 2021

- Added: IMAP processing now traps "NO" and "BAD" responses during login.
- Change: Removed limits on the number & size of saved subject lines that can be returned by seeGetHeader()
- Change: Allows ISO & UTF subjects to be broken in mid-line.
- Change: SEE_REPLACE_UNDERSCORES (EnableRFC2047) default changed to TRUE
- Added: Added support for OAUTH2 - user MUST provide access token !
- Fixed: Base64 encoded passwords now replaced with astericks in log file (default)
- Change: Increased password buffer to 256 characters (supports SendGrid passwords)

Version 8.3: April 20, 2022

- Fixed problem when renaming attachments "on the fly" [oldname|newname].
- Added renaming of inline images "on the fly" [oldname|newname].
- Write authentication protocol selected by user to the log file.
- Added "Unexpected empty string" (SEE_EMPTY_STRING) error code.
- Allow "From:" header to be split over multiple lines.
- Protocol XOAUTH2 must be explicitly enabled - seeIntegerParam(0, SEE_AUTHENTICATE_PROTOCOL, AUTHENTICATE_XOAUTH2).

Version 8.4: Marh 1, 2023

- Message-ID header written to all outgoing email.
- Added diagnostics to decoding UTF-8 filenames.
- Fixed problem if SSL config string did not end with LF.
- Increased OAUTH2 password buffer to 2048 bytes.
- Added function seeMakeXOAuth2() that creates OAUTH2 password string.

Version 8.5: November 7, 2024

- Changed MakeMessageID() so that it extracts from the last email address
- Fixed problem with initializing HandleTable between calls to seeAttach.
- Check for embedded spaces in passwords.
- Can start multiple copies of Stunnel.
- Streamlined termination process when file could not be created or opened.
- Recognizes Windows-1256 parts (Arabic text).