SMTP/POP3/IMAP Email Engine

Library for COBOL

Programmer's Manual

(SEE4CB)

Version 7.3

December 23, 2014

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 COBOL (SEE4CB**) library is a toolkit that allows software developers to quickly add SMTP and POP3/IMAP capabilities to their COBOL applications.

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 COBOL programming environment.

The **SMTP/POP3/IMAP Email Engine for COBOL** component library works with Win32 Fujitsu COBOL and any other COBOL (such as Micro Focus COBOL) capable of calling the Windows Win32 API. **SEE4CB** includes numerous example programs that demonstrate SMTP and POP3/IMAP functions used to build mail applications using the SEE4CB library.

A Win32 DLL (Dynamic Link Library) is provided. **SEE4CB** runs under all versions of Windows (Windows 95, Windows 98, Windows ME, Windows 2000, Windows 2003-2012, Windows NT, Windows XP, Windows Vista, Windows 7 and Windows 8). A Win64 DLL is available for Win64 programming languages. 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, PowerBASIC, Xbase++, dBase, etc.) capable of calling the Windows API.

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

- SEE4CB is a standard Windows DLL and is much smaller than an OCX or ActiveX control.
- SEE4CB does NOT depend on ActiveX or similar "support" libraries.
- SEE4CB is fully threadable.
- The SEE4CB 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), Visual Basic (SEE4VB), Delphi (SEE4D), PowerBASIC (SEE4PB), Visual FoxPro (SEE4FP), 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 computer 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 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 server including MIME attachments.
- Remove contents of attachments before writing to disk.
- Forward Email.
- Decode email from a file
- Run up to 64 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.
- Supports CHARSET_WIN_1250, CHARSET_WIN_1252 and CHARSET_WIN_1255.
- 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.
- Use with GMAIL/Yahoo/Hotmail servers requiring SSL with (free) STUNNEL proxy server.
- Start and terminate external programs from within an application.
- Is fully thread safe
  .
- Implemented as a **standard** Windows DLL, which will work with all versions of Windows.
- Includes 32-bit (SEE32.DLL) DLL. A SEE64.DLL is available.
- Works with Win 32 Fujitsu COBOL and any other COBOL (such as Micro Focus COBOL) capable of calling the Windows Win32 API
- Does **not** depend on support libraries. Makes calls to core Windows API functions only.
- Works with Windows 95/98/Me/NT/2000/2003/2008/2012/XP/Vista/Windows 7/Windows 8.
- License covers all programming languages.
- 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, Visual Basic, VB.NET, Access and Excel.
- Is compatible with “Managed Code”.
- The evaluation version is fully functional.
- Royalty free distribution (no run-time fees) with your compiled application

Registration includes one year of technical support and free downloadable updates.

A good selection of COBOL 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.
 BCAST       :  Sends bulk email to one recipient per email.
 FORWARD     :  Forwards undecoded email.
 GETRAW      :  Downloads specified email without decoding.
 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 “direct” 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
                 “direct method.”
 ReadSSL     :  Downloads email from POP3 server requiring SSL.
 STATUS      :  Lists all email on server and displays DATE, FROM and SUBJECT
                header fields.
 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 Set

The complete set of documentation consists of three manuals in Adobe PDF format. This is the first manual (SEE_4CB) in the set.

- SEE4CB Programmer's Manual (`SEE_4CB.PDF`)
- SEE User's Manual (`SEE_USR.PDF`)
- SEE Reference Manual (`SEE_REF.PDF`)

The SEE_4CB Programmer's Manual is the language specific (COBOL) manual. All COBOL 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 COBOL** product page at:

`http://www.marshallsoft.com/see4cb.htm`

1.3 Example Program

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

```
*> attach SEE
CALL "seeAttach" WITH STDCALL USING
   BY VALUE     NBR-CHANS     *> number of channels
   BY VALUE     SEE-KEY-CODE  *> See KEYCODE.CBI
END-CALL.
*> connect to SMTP server
DISPLAY "Calling seeConnectTo()..."
CALL "seeConnectTo" WITH STDCALL USING
   BY VALUE     IS-ZERO       *> channel
   BY REFERENCE SMTP-SERVER   *> SMTP server
   BY REFERENCE EMAIL-FROM    *> sender's email address
   BY REFERENCE IS-ZERO       *> no Reply-To header
END-CALL.
*> send email
DISPLAY "Calling seeSendEmail()..."
CALL "seeSendEmail" WITH STDCALL USING
   BY VALUE     IS-ZERO       *> channel
   BY REFERENCE EMAIL-TO      *> recipient
   BY REFERENCE IS-ZERO       *> no CC list
   BY REFERENCE IS-ZERO       *> no BCC list
   BY REFERENCE EMAIL-SUBJ    *> email subject
   BY REFERENCE EMAIL-MSG     *> text of message
   BY REFERENCE IS-ZERO       *> no attachment
END-CALL.
*> close and release SEE
CALL "seeClose" WITH STDCALL USING
   BY VALUE     IS-ZERO       *> channel
END-CALL.
CALL "seeRelease" WITH STDCALL
END-CALL.
```

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 SEE4CB, your COBOL compiler should already be installed on your system and tested.

(2) Unzip SEE4CB73.ZIP (or SEExxxx.ZIP where xxxx is your Customer ID) using any Windows unzip program.

(3) Run the installation program, SETUP.EXE, which will install all SEE4CB files, including copying SEE32.DLL to your Windows directory.

(4) Compile and run SEEVER.CBL to test your installation. The remaining example programs must be edited with your email parameters before compiling.

1.5 Uninstalling

First, run UINSTALL.BAT, which will delete SEE32.DLL from your Windows directory, typically C:\WINDOWS for Windows 95/98//2003/2012/Me/XP/Vista/Win7/Win8 or C:\WINNT for Windows NT/2000.

Second, delete the SEE project directory created when installing SEE4CB.

1.6 Pricing

A developer license for the SMTP/POP3/IMAP Email Library can be purchased for $115 USD. Purchasing details can be found in the SEE User's Manual, Section 1.4, "How to Purchase", (/docs/see_usr.pdf or `http:/www.marshallsoft.com/see_usr.pdf`).

Also see INVOICE.TXT provided with the evaluation version or order directly on our web site at

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

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

1.7 Updates

When a developer license is purchased, you will be sent a set of registered DLLs plus a license file (SEExxxx.LIC, where xxxx is your Customer ID). The license file can be used to update the registered DLL's for a period of one year 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:

- $30 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).
- $75 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 /SEE4CB/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/2003-2012/Vista/Windows 7/Windows 8 and Windows NT/2000.

The SEE4CB library has been tested with Fujitsu COBOL and also works with Micro Focus COBOL

The SETUP installation program will copy the SEE32.DLL to the Windows directory and copies the SEE4CB files to the directory specified (default \SEE4CB). Three sub-directories are created, as follows:

```
DOCS – All documentation files
APPS – All example code
DLLS – All DLL’s
```

2.1 Dynamic Link Libraries

The **SMTP/POP3/IMAP Email** component library includes both a Win64 and 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.CC. The keycode for the evaluation (shareware) version is 0. You will receive a new keycode and a set of new DLL's when registering. 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 and SEE16.DLL from the Windows search path or delete them.

2.3 Win32 STDCALL and DECLSPEC

SEE32 is written in ANSI C and is compiled using the _stdcall and _declspec keywords. This means that SEE4CB 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.

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 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.

3 Compiler Issues

3.1 Versions of Win32 COBOL

The email functions in SEE32.DLL can be called from any program which is capable of calling the Windows Win32 API. In order to call the email functions, the proper syntax for calling external Windows DLL's must be used. This syntax is different for each COBOL compiler.

For Fujitsu, the function interface file is SEE32.CBI. For Micro Focus, the interface file is SEE32.MFI.

3.2 Compiling Fujitsu COBOL Programs

The Fujitsu COBOL Project Manager is used to compile and link the example programs. When linking, remember to include SEE32.LIB in the list of files to link.

Alternatively, the FCL.BAT file can be used to invoke the Fujitsu compiler and linker from the command line. Be sure to copy COBOL.CBI from the Fujitsu COBOL97 directory to the directory which contains the example programs.

```
FCL AUTHEN
FCL BCAST
FCL GETRAW
FCL HTML
FCL MAILER
FCL MAILER2
FCL READER
FCL READER2
FCL SEEVER
FCL STATUS
FCL MPARTS
FCL FORWARD
FCL POP3RD
FCL MailSSL
FCL ReadSSL
```

3.3 Compiling Micro Focus COBOL Programs

We do not have Micro Focus COBOL, although we have many customers who use SEE4CB with it.

In order to compile for MicroFocus COBOL, examine the Micro Focus version of the SEEVER program (SEEVERMF.CBL). If you would like the complete example set for MicroFocus COBOL, send an email to support [at] marshallsoft.com, and we will be glad to send it to you.

3.4 Other Win32 COBOL Programming Languages.

If you develop an interface file for other COBOL languages, we would really appreciate a copy. Email the interface file, along with any example programs that you have converted, to support [at] marshallsoft.com.

If you are programming in another Win32 COBOL language, we may already have the necessary interface file. Let use know which COBOL you want to use.

4 Example Programs

Each example program, with the exception of SERVER.CBL, must be edited with your email parameters before compiling. For example, to change the "mail.hiwaay.net" below to "mail.mine.com", the following changes must be made:

```
     000360  01  SMTP-SERVER.
     000370      05  FILLER PIC X(15) VALUE "mail.hiwaay.net".
     000380      05  FILLER PIC X VALUE X'00'.
```

to

```
     000360  01  SMTP-SERVER.
     000370      05  FILLER PIC X(13) VALUE "mail.mine.net".
     000380      05  FILLER PIC X VALUE X'00'.
```

Note that the length was changed from 15 characters to 13 to accommodate the shorter server name.

Before writing your own programs, compile and run several of the example programs. SMTP programs send email. POP3 programs read email.

4.1 Connectionless Example Programs

The SEEVER example program does not require a connection to an SMTP server.

4.1.1 SEEVER

The first example program SEEVER displays the SEE library version number and registration string. It does not require a connection. See SEEVER.CBL for Fujitsu or SEEVERMF.CBL for MicroFocus.

Compile and run SEEVER first. It will verify that SEE32.DLL can be found at runtime and that your keycode is specified correctly.

4.2 SMTP Example Programs

There are eight SMTP email example programs. SMTP programs send email.

4.2.1 AUTHEN

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

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

4.2.2 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 the SMTP server and your email address, the file containing the email message to send must be created along with another file containing the list of recipients. See EMAIL.LST for an example.

BCAST.CBL must be edited with your email parameters before compiling.

4.2.3 Forward

The FORWARD example program forwards an (undecoded) email to a new recipient.

FORWARD.CBL must be edited with your email parameters before compiling.

4.2.4 HTML

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

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

4.2.5 ISO8859

The ISO8859 example program sends a text message and subject line that are ISO-8859-1encoded in Italian. 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).

ISO8859.CBL must be edited with your email parameters before compiling.

4.2.6 MAILER

The MAILER example program emails a message, including an optional MIME attachment. The MAILER.CBL program must be edited with your email parameters before compiling. Note that <> brackets are required around all email addresses.

MAILER.CBL must be edited with your email parameters before compiling.

4.2.7 MAILER2

The MAILER2 example program operates like MAILER, except that it runs in direct mode rather than indirect mode like MAILER.CBL. The primary use of direct mode is to return control to the calling application code as the email is being sent for the purpose of displaying progress. Refer to the SMTP/POP3/IMAP User's Manual, Section 7.2, "Indirect Method" (/DOCS/SEE_USR.PDF or online at (www.marshallsoft.com/see_usr.pdf).

The MAILER2.CBL example program must be edited with your email parameters before compiling.

4.2.8 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 SEE4CB\DOCS directory or `http://www.marshallsoft.com/stunnel.htm`.

The MailSSL.CBL program must be edited with your email parameters before compiling.

4.2.9 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).

The Mparts.CBL program must be edited with your email parameters before compiling.

4.2.10 Authen.cbi (code segment)

This code segment can be inserted into any SMTP program that requires SMTP Authentication.

4.2.11 UsePort587.cbi (code segment)

This code segment can be inserted into any SMTP program that requires port 587 rather than the default port 25.

4.3 POP3 Example Programs

There are four POP3 email example programs. These examples read email.

4.3.1 GETRAW

The GETRAW example program downloads a specified email message without decoding it. This program is used to see what the email looks like on the server.

The GETRAW.CBL program must be edited with your email parameters before compiling.

4.3.2 POP3RD

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

The POP3RD.CBL program must be edited with your email parameters before compiling.

4.3.3 READER

The READER example program can read email, including multiple MIME attachments, from your POP3 email server, deleting each email after being read.

The READER.CBL program must be edited with your email parameters before compiling.

4.3.4 READER2

The READER2 example program operates like READER, except that it runs in direct mode rather than indirect mode like READER.CBL. The READER2.CBL program must be edited with your email parameters before compiling.

The READER2.CBL program must be edited with your email parameters before compiling.

4.3.5 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 \SEE4CB\DOCS directory or `http://www.marshallsoft.com/stunnel.htm`.

The ReadSSL.CBL program must be edited with your email parameters before compiling.

4.3.6 STATUS

The STATUS example program reads the number of email messages waiting on your POP3 email server, and displays the "DATE:", "FROM:", and "SUBJECT:" header fields from each email.

The STATUS.CBL program 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 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]
```

The ImapFlag.CBL program must be edited with your email parameters before compiling.

4.4.2 ImapSrch

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
```

The ImapSrch.CBL program must be edited with your email parameters before compiling.

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, COBOL, PowerBASIC, FoxPro, dBase, Xbase++, COBOL, and Fortran) use the same identical DLL.

Version 1.0: June 8, 1998.

- Initial release

Version 2.0: [COBOL version not released]

Version 2.1: [COBOL version not released]

Version 3.0: [COBOL version not released]

Version 3.1: September 1, 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.

Version 3.2: February 28, 2000.

- 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, READER2, and MAILER2 example programs.
- Added seeCommand function.

Version 3.3: December 11, 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 25, 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 example program.

Version 3.6: May 5, 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 18, 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 and MParts example programs.

Version 4.0: August 23, 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: June 6, 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 17, 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)
- Consider TAB's as ASCII characters (TABIsASCII=1) [bld 6]
- 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.

Version 5.2: May 4, 2010 (Win32) version

- 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 SEE_GUT_ATTACHMENTS to seeIntegerParam to remove contents of incoming attachments.

Version 6.0: March 25, 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: December 2, 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
- NAME_BUF increased from 1024 to 5120 characters.
- 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 18, 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: October 4, 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 23, 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)

[END]