SMTP/POP3/IMAP Email Engine Library for Visual Basic

Programmer's Manual

(SEE4VB)

Version 8.0

October, 2018

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 Visual Basic** (**SEE4VB**) library is a toolkit that allows software developers to quickly develop SMTP and POP3/IMAP email applications in Visual Basic, Visual Studio .NET (VB.NET) or VBA (Visual Basic for Applications).

The **SMTP/POP3/IMAP Email Engine** (SEE) is a component DLL library 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, 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 Programmer's Manual for Visual Basic** provides information needed to compile and run programs in a Visual Basic programming environment.

The **SMTP/POP3/IMAP Email Engine for Visual Basic** component library supports and has been tested with all versions of Microsoft Visual Basic (VB 4.0 – VB 6.0), Microsoft Visual Studio .NET Framework and Microsoft Visual Studio through Visual Studio 2013. **SEE4VB** can also be used with any VBA (Visual Basic for Applications) language such as Excel, Access, MS Office, etc. as well as with PowerBuilder.

SEE4VB includes numerous example programs that demonstrate SMTP and POP3/IMAP email functions used to create software applications using the **SEE4VB** library. All examples compile with either VB-4.0 through VB-6.0 or with Visual Studio. Examples for Visual Basic for Applications (VBA) are also provided.

SEE4VB runs under all 32-bit and 64-bit versions of Windows through Windows 10. The **SMTP/POP3/IMAP Email Engine SDK** DLLs (SEE32.DLL and SEE64.DLL) can also be used from any language (C/C++, Delphi, Visual FoxPro, COBOL, Xbase++, dBase, PowerBASIC, etc.) capable of calling the Windows API.

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

- SEE is a standard Windows DLL (NOT an OCX or ActiveX control) and is much smaller than a comparable OCX or ActiveX control.
- SEE does NOT depend on ActiveX or similar "support" libraries.
- SEE is fully thread-safe.
- SEE 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), Xbase++ (SEE4XB) and Cobol (SEE4CB). All versions of the **SEE** library use the same DLLs (SEE32.DLL and 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)** component 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 mail 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.
- Decode email from a file
- Use with GMAIL/Yahoo/Live Mail servers requiring SSL/TLS.
- Start and terminate external programs from within an application.
- Run up to 32 independent 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
- Construct non-standard email that must be quoted (ex., EDIFACT)
- Includes 65 functions for SMTP and POP3 mail control.
- Dozens of switches provided to control how email is sent or received.
- Supports setting priority via X-Priority header field.
- Removes contents of attachments before writing to disk.
- 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.
- Includes Win32 and Win64 DLLs.
- Supports all versions of Visual Basic, from VB 4.0 through Visual Studio 2013.
- Works with Microsoft Visual Studio .NET Framework.
- Works with PowerBuilder.
- Works with Visual Basic for Applications (VBA) such as Excel, Access, and Microsoft Office
- 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, Microsoft Office, PowerBuilder, Access and Excel.
- Is compatible with "Managed Code".
- Supports 32-bit and 64-bit Windows through Windows 10.
- License covers all programming languages.
- Royalty free distribution with your compiled application.
- Free technical support and downloadable updates for one year.
- Online documentation as well as in printable format.
- Evaluation versions are fully functional. No unlock code is required.

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

[PROGRAM]		[DESCRIPTION]
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 email header information.
GB2312	:	Sends email that is GB2312 (simplified Chinese).
GetRaw	:	Downloads specified email without decoding.
HTML	:	Sends html encoded email with attachments.
ISO8859	:	Sends ISO-8859 encoded message and subject line
ImapFlagsT	:	Tests manipulation of flaps on IMAP server.
ImapMBTest	:	Tests IMAP functions.
ImapSEARCH	:	Tests IMAP search capability.
MailSSL	:	Sends email (SMTP server requires SSL).
Mailer	:	Sends email with optional attachment.
Mparts	:	Sends multipart MIME email.
POP3Rd	:	Specifies email message file to read and decode.
ReadSSL	:	Downloads email (POP3 server requires SSL).
Reader	:	Downloads email & attachments and saves to disk.
SeeVer	:	Displays SEE Version/Build number and registration string.
Status	:	Displays email header information.
TestConn	:	Tests connection to a specified server and port.
vbaMail	:	VBA module that sends email.
vbaMailSSL	:	VBA module that sends email (SMTP server requires SSL).
vbaRead	:	VBA module that reads email.
vbaReadSSL	:	VBA module that reads email (POP3 server requires SSL).
vbaStatus	:	VBA module displays selected headers.

1.2 Documentation Set

The complete set of documentation consists of three manuals in Adobe PDF format. This is the first manual (SEE_4VB) in the set.

- SEE4VB Programmer's Manual (SEE_4VB.PDF)
- SEE User's Manual (SEE_USR.PDF)
- SEE Reference Manual (SEE_REF.PDF)

The SEE_4VB Programmer's Manual (SEE_4VB.PDF) is the language specific (Visual Basic) manual. All Visual Basic programming issues such as compiling, compilers and example programs are discussed in this manual.

The SEE User's Manual (SEE_USR) discusses SMTP and POP3/IMAP email processing as well as language independent programming issues such as application notes. Purchasing and licensing information is also provided.

The SEE Reference Manual (SEE_REF) contains details on each individual SEE function and provides a list of SEE error codes.

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

http://www.marshallsoft.com/see4vb.htm

1.3 Example Program

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

```
Dim Code As Integer
Dim Server, From As String
Dim ToList, CClist, BCClist As String
Dim Subject, Message, Attachments As String
IsNull  = Chr(0)
Server  = "mail.yourisp.com
From = "my name<me@myisp.com>"
ToList   = "<support@marshallsoft.com>"
CClist = Chr(0)
BCClist = Chr(0)
Subject = "Visual Basic Test"
Message = "Emailed from SEE4VB!"
Code = seeAttach(1, SEE_KEY_CODE)              ' keycode is 0
If Code < 0 Then
  ' error calling seeAttach !
  '. . .
End If
  ' connect to SMTP mail server
Code = seeSmtpConnect(0, Server, From, From)   ' chan, server, From & Reply-To addr
If Code < 0 Then
  ' error calling seeSmtpConnect !
  '. . .
End If
` send the mail
Code = seeSendEmail(0, ToList, CClist, BCClist, Subject, Message, Attachments)
If Code < 0 Then
  ' error calling seeSendEmail !
  '. . .
End If
' close connection to server
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 SEE4VB, your Visual Basic compiler (any version) should already be installed on your system and tested.

(2) Unzip SEE4VB80.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 SEE4VB files, including copying SEE32.DLL and SEE64.DLL to the Windows directory.

VB 4.0, VB 5.0, and VB 6.0 project filenames end with the extension ".vbp", and VB.NET/Visual Studio project filenames end with "vbproj". For example,

```
SEEVER.VBP    --- Project file for 32-bit Visual Basic (VB_4.0,5.0,6.0)
SEEVER.VBPROJ --- Project file for VB.NET / Visual Studio
```

Note that no DLL registration is required.

1.5 Uninstalling

Uninstalling SEE4VB is very easy.

First, run UINSTALL.BAT, which will delete SEE32.DLLand SEE64.DLL from the Windows directory, typically C:\WINDOWS.

Second, delete the SEE project directory created when installing SEE4VB.

1.6 Pricing

A developer license for the SMTP/POP3/IMAP Email Library can be registered for $119 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
https://www.marshallsoft.com/vmorder.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 set of registered DLLs plus a license file (SEExxxx.LIC). 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, your license must be updated if you want to be able to download updates. The license can be updated for $33 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, updates are $77.

2 Library Overview

The **SMTP/POP3/IMAP Email** component library has been tested on multiple computers running Windows XP through Windows 10.

The SEE4VB library has been tested with several Visual Basic compilers, from VB 4.0 through VB 6.0, Microsoft Visual Basic .NET (**VB.Net**.) and Microsoft Visual Studio (2003 through 2013). SEE can also be used with any VBA language such as Excel, Access, MS Office as well as PowerBuilder.

The SETUP installation program will copy the DLL's to the Windows directory. Refer to Section 1.4 "Installation".

After SETUP is run, the SEE4VB files are copied to the directory specified (default \SEE4VB). 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 SDK uses a Win32 [SEE32.DLL] and Win64 [SEE64.DLL] 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".

The following files can be found in the DLL sub-directory when SETUP is run:

```
see32.dll   - Win32 version of SEE
see64.dll   - Win64 version of SEE
```

2.2 Keycode

SEE32.DLL and SEE64.DLL are encoded with a keycode. The keycode is a 9 or 10 digit decimal number (unless it is 0), and will be found in the file KEYCODE.BAS (and KEYCODE.VB). The keycode for the evaluation version is 0. The developer will receive a new keycode and SEE32.DLL after 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 SEE64.DLL from the Windows search path or delete them.

2.3 Win32 / Win64 STDCALL and DECLSPEC

Our libraries are written in ANSI C and compiled using the _stdcall and _declspec keywords. This means that SEE4VB uses the same calling conventions and file naming conventions as the Windows API

The SEE library 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

SEE4VB (SEE32.DLL and SEE64.DLL) is thread safe and can be used from any Windows application capable of using threads.

2.5 Dynamic Strings

2.5.1 Passing VB Strings to DLL Functions

When passing a string to a DLL function, the DLL looks for a null character to terminate the string. This null character CHR(0) is normally present in Visual Basic strings, but can be lost if the string is modified by Visual Basic at runtime. This problem can be overcome by appending CHR(0) to the end of strings passed to **SEE** functions.

2.5.2 Passing VB String Buffers to DLL Functions.

The Visual Basic language uses a technique known as "garbage collection" to manage string space at runtime, and may be called internally at any time by the Visual Basic runtime, asynchronous to what you may be doing in your code.

When passing a string buffer to a DLL function into which text will be copied, it is strongly recommended that the local string be allocated immediately before use. For example, a string buffer is passed to the dllGetMessage function, which copies a text message into it. Note that `SPACE(80)` is called immediately before dllGetMessage.

```
Dim Code As Integer
Dim Buffer As String * 80
' allocate buffer just before call to dllGetMessage
Buffer = SPACE(80)
' copy message into 'Buffer'
Code = dllGetMessage(Buffer, 80)
' message text is now in 'Buffer'
```

This technique is not necessary for passing a string to a DLL function, only when passing a buffer to a DLL into which data is to be placed by the DLL function.

2.6 Visual Studio (VB.NET)

SEE4VB works with all versions of Visual Studio through VS 2013.

There are a few differences between VB 4/5/6 and Visual Studio (VB.NET) that affect writing programs that use the **SEE** library.

1. Variables that are declared "As Long" in VB 4/5/6 are declared "As Integer" in Visual Studio (VB.NET).

2. Fixed length strings are not supported in Visual Studio (VB.NET). When calling any **SEE** function that can return a string, memory for the string variable must be allocated first. For example:

```
Buffer = Space(128)
Length = seeErrorText(0, ErrCode, Buffer, 128)
```

3. Some Visual Basic functions must be fully qualified. For example, instead of LEFT, use `Microsoft.VisualBasic.Left`

4. The module SEE32.VB (not SEE32.BAS) must be included in all VB.NET programs. For Win64 programs, include SEE64.VB instead.

2.7 Visual Basic for Applications (VBA)

The **SMTP/POP3/IMAP Email** component library can be used with Microsoft VBA applications such as ACCESS, EXCEL, and WORD. The VBA example modules are vbaMail, vbaMailSSL, vbaRead, vbaReadSSL, and vbaStatus. The file vbaDeclare.bas contains all constants and function declarations for using SEE4VB in VBA applications.

2.7.1 ACCESS

1. Start ACCESS
2. Open a table
3. Click "Database Tools" then double click "Visual Basic".
4. In the "Microsoft Visual Basic for Applications" window, click "Insert" then "Module"
5. Paste the VBA code.
6. Click "Run" on the VBA menu, then "Run Sub"
7. When the "Macros" menu is displayed, enter the VBA coder name then CR.

2.7.2 EXCEL and WORD

If "Developer" is not displayed on the ribbon line:

1. Start EXCEL (or WORD)
2. Click "File" then "Options"
3. Click "Customize Ribbon" in " Options" window.
4. Click "Macros"
5. Check "Developer"
6. Click "OK"

To add VBA code:

1. Start EXCEL or WORD)
2. Click "Developer"
3. Double-click "Visual Basic".
4. In the "Microsoft Visual Basic for Applications" window, click "Insert" then "Module"
5. Paste the VBA code.
6. Click "Run" on the VBA menu, then "Run Sub"
7. When the "Macros" menu is displayed, enter the VBA coder name then CR.

2.8 Power Builder

SMTP/POP3/IMAP Email Engine can also be used with Power Builder applications. Refer to PBUILDER.TXT in the \APPS subdirectory for more information.

`SEE32.PBI` : Power Builder declaration file.

2.9 Adding SEE4VB to a Project

Copy SEE32.BAS (if running VB_4.0 /5/6), SEE32.VB (if running Visual Studio Win32), or SEE64.VB (if running Visual Studio Win64) to the same directory (folder) as the application program to which you want to add **SEE** code. You will find these files in the \APPS sub-directory (folder) created when you ran SETUP

After the project settings are modified (see Sections 2.09.1 and 2.09.2 below), the first **SEE** function that must be called is **seeAttach**. Often, this is best done when the Visual Basic form is first loaded. For example,

```
Private Sub Form_Load()
Dim Code As Integer
Code = seeAttach(1, SEE_KEY_CODE)
If Code < 0 Then
  MsgBox "ERROR: Cannot attach. Check SEE_KEY_CODE."
  End
End If
End Sub
```

The last **SEE** function that must be called is **seeRelease**. This is often best done just before the application terminates. If there is an EXIT button on the form, "Code = seeRelease()" can be added. For example,

```
Private Sub Exit_Click()
Dim Code As Integer
Code = seeRelease()
End
End Sub
```

2.9.1 Adding SEE4VB to aVB_4.0 , 5.0, or 6.0 Project

Open the existing project with "File", "Open Project". Then choose "Insert", "Module", then add SEE32.BAS and KEYCODE.BAS to your project. If prompted to add "DAO 2.50 Object Library", choose "no".

SEE functions can now be called from your Visual Basic program.

2.9.2 Adding SEE4VB to your VB.NET Project

Open the existing project with "File", "Open Project". Then choose "Project", "Add Module", then add SEE32.VB and KEYCODE.VB to your project.

SEE functions can now be called from your VB.NET program.

2.9.3 Adding SEE4VB to your VS2005 through VS2013 Project

Open the existing project with "File", "Open Project". Then choose "Project", "Add Module", then add modules SEE32.VB (or SEE64.VB) and KEYCODE.VB to your project. If empty modules are created, replace the contents of these modules with the contents of the modules of the same name provided by us.

SEE functions can now be called from your Visual Studio 2005, 2008, 2010, 2012, or 2013 VB program.

2.10 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 refer to the file **seeErrors.txt** for a list of all Winsock and SEE error codes.

3 Compiler Issues

The **SMTP/POP3/IMAP Email Engine for Visual Basic** component library supports and has been tested with all versions of Microsoft Visual Basic including:

- Visual Basic 4, 5, 6
- Visual Basic .NET (VB 7)
- Visual Basic .NET 2003 (VB 7.1)
- Visual Studio 2005
- Visual Studio 2008
- Visual Studio 2010
- Visual Studio 2012
- Visual Studio 2013
- Visual Studio 2015

SEE can also be used with any VBA language such as Excel, Access, MS Office as well as PowerBuilder.

3.1 Visual Basic Makefiles

The first Visual Basic for Windows (VB Version 3.0) uses a text file known as a "Visual Basic makefile" `(.MAK)` to list all the file components necessary to compile a program. Beginning with Visual Basic Version 4.0, the "Visual Basic Project file" `(.VBP)` was added. Both formats are "project files".

Beginning with Visual Studio 2003 (VB.Net), Visual Basic project files use extension ".vbproj".

3.2 Compiling Programs

Project files are provided for all examples. Project files for VB 4/5/6 end with the extension ".vbp" and end with ".vbproj" for Visual Studio (and VB.NET).

The example programs can be compiled from the Visual Basic development environment using the provided Visual Basic project files. Choose "File", then "Open Project" from the main VB menu.

After opening a project, VB v5.0 (and VB v6.0) users can save the project files in the VB v5.0 (or VB v6.0) format. When saving the example programs in VB v5.0 or VB v6.0 format, answer "no" if asked to add the "Microsoft DAO v2.5 library".

Compile and run SEEVER32 as the first example to check your installation. SEEVER does not require a TCP/IP connection.

3.3 Explicitly Loading SEE32.DLL / SEE64.DLL

When an application program runs that makes calls to SEE32.DLL the Windows operating system will locate SEE32.DLL by searching the directories as specified by the Windows search path. If the SEE32.DLL is placed in the \WINDOWS directory, it will always be found by Windows.

SEE32.DLL can be loaded from an explicit location by replacing "SEE32.DLL" in SEE32.BAS or SEE32.VB by the full path. For example, to load SEE32.DLL from C:\SEE4VB\APPS, the first entry in SEE32.VB would be:

```
Declare Function seeAbort Lib "C:\SEE4VB\APPS\SEE32.DLL"(ByVal Chan As Integer)
As Integer
```

The above also applies to SEE64.DLL as well as SEE32.DLL

4 Example Programs

Multiple Visual Basic example programs are included in **SMTP/POP3/IMAP Email Engine for Visual Basic** (**SEE4VB**). Examples for both Visual Basic 4/5/6 and Visual Studio (and VB.NET) are included.

Each example program comes with a VB project file.

- VB 4/5/6 project files end with ".VBP "
- Visual Studio / VB.Net project files end with ".VBPROJ".

Before writing your own programs, compile and run several of the example programs.

4.1 Connectionless Example Programs

Several 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. Its purpose is to display the **SE**E version, build, and registration string as well as to verify that SEE32.DLL or SEE64.DLL is being found and loaded by Windows.

```
Open project SeeVer.VBP              -- VB 4, 5, 6
Open project SeeVer.vbproj           -- Visual Studio (VB.NET)
Open project SeeVer(VS2008).vbproj   -- Visual Studio 2008
Open project SeeVer(VS2010).vbproj   -- Visual Studio 2010
Open project SeeVer(VS2012).vbproj   -- Visual Studio 2012
Open project SeeVer(VS2013).vbproj   -- Visual Studio 2013
```

4.1.2 CodeTest

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

```
Open project CodeTest.vbp              -- VB 4, 5, 6
Open project CodeTest.vbproj           -- Visual Studio (VB.NET)
Open project CodeTest(VS2008).vbproj   -- Visual Studio 2008
Open project CodeTest(VS2010).vbproj   -- Visual Studio 2010
Open project CodeTest(VS2012).vbproj   -- Visual Studio 2012
Open project CodeTest(VS2013).vbproj   -- Visual Studio 2013
```

4.1.3 Pop3Read

The Pop3Read example program uses the **seePop3Source** function to specify an (undecoded) email message file to be read and decoded.

```
Open project POP3Read.vbp            -- VB 4, 5, 6
```

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.

```
Open project TestConn.vbproj           -- Visual Studio (VB.NET)
Open project TestConn(VS2008).vbproj   -- Visual Studio 2008
Open project TestConn(VS2010).vbproj   -- Visual Studio 2010
Open project TestConn(VS2012).vbproj   -- Visual Studio 2012
Open project TestConn(VS2013).vbproj   -- Visual Studio 2013
```

4.2 SMTP Email Example Programs

There are ten 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 (ESMTP) server using SMTP Authentication. You must connect to an SMTP server that allows authentication.

```
Open project Authen.vbp              -- VB 4, 5, 6
```

4.2.2 Auto

AUTO ("auto-responder") uses two channels to read email on the POP3 server and at the same time automatically responds to all new email using the SMTP server. Edit your TCP/IP parameters in AUTO.FRM (line 73) before compiling.

```
Open project Auto.vbp                -- VB 4, 5, 6
```

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.

```
Open project BCast.vbp               -- VB 4, 5, 6
```

4.2.4 Forward

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

```
Open project Forward.vbp             -- VB 4, 5, 6
```

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 Microsoft OutLook).

```
Open project GB2312.vbp              -- VB 4, 5, 6
```

4.2.6 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.

```
Open project MailSSL.VBP              -- VB 4, 5, 6
Open project MailSSL.vbproj           -- Visual Studio (VB.NET)
Open project MailSSL(VS2008).vbproj   -- Visual Studio 2008
Open project MailSSL(VS2010).vbproj   -- Visual Studio 2010
Open project MailSSL(VS2012).vbproj   -- Visual Studio 2012
```

4.2.7 HTML

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

```
Open project HTML.vbp                 -- VB 4, 5, 6
Open project HTML.vbproj              -- Visual Studio (VB.NET)
Open project HTML(VS2008).vbproj      -- Visual Studio 2008
Open project HTML(VS2010).vbproj      -- Visual Studio 2010
Open project HTML(VS2012).vbproj      -- Visual Studio 2012
Open project HTML(VS2013).vbproj      -- Visual Studio 2013
```

4.2.8 ISO8859

The ISO8859 example program sends a text message and subject line that is 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 Microsoft OutLook).

```
Open project ISO8859.vbp               -- VB 4,5,6.
```

4.2.9 Mailer

MAILER emails a message, including optional MIME attachments. All required parameters are input using a dialog box at runtime. Note that <> brackets are required around the email address.

```
Open project Mailer.vbp                 -- VB 4, 5, 6
Open project Mailer.vbproj              -- Visual Studio (VB.NET)
Open project Mailer(VS2008).vbproj      -- Visual Studio 2008
Open project Mailer(VS2010).vbproj      -- Visual Studio 2010
Open project Mailer(VS2012).vbproj      -- Visual Studio 2012
Open project Mailer(VS2013).vbproj      -- Visual Studio 2013
```

4.2.10 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 a PDF file (.pdf).

```
Open project MParts.vbp               -- VB 4, 5, 6
```

4.3 POP3/IMAP Email Example Programs

There are six POP3 email example programs. These examples read email using the POP3 server.

All of these examples work with IMAP servers by defining the symbol CONNECT_TO_IMAP_SERVER in the program source code before compiling.

4.3.1 Auto

AUTO ("auto-responder") uses two channels concurrently to read email on the POP3 server and at the same time automatically respond to all new email using the SMTP server. Edit your TCP/IP parameters in AUTO.FRM (line 73) before compiling.

```
Open project Auto.vbp                -- VB 4, 5, 6
```

4.3.2 GetRaw

GETRAW is a Win32 program that downloads a specified email message without decoding it. This is used to see what the email looks like on the server. Also see the READER example program, which can also download email without decoding it.

All required parameters are coded inside GETRAW.FRM (line 67) and must be edited before compiling.

```
Open project GetRaw.vbp              -- VB 4, 5, 6
```

4.3.4 Pop3Read

The Pop3Read example program uses the **seePop3Source** function to specify an (undecoded) email message file to be read and decoded.

```
Open project POP3Read.vbp            -- VB 4, 5, 6
```

4.3.5 Reader

READER can read email, including multiple MIME attachments, from a POP3/IMAP server, optionally deleting each email after being read. READER can also download email without decoding. All required parameters are input at runtime.

```
Open project Reader.vbp              -- VB 4, 5, 6
Open project Reader.vbproj           -- Visual Studio (VB.NET)
Open project Reader(VS2008).vbproj   -- Visual Studio 2008
Open project Reader(VS2010).vbproj   -- Visual Studio 2010
Open project Reader(VS2012).vbproj   -- Visual Studio 2012
Open project Reader(VS2013).vbproj   -- Visual Studio 2013
```

4.3.6 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. All required parameters are input at runtime.

```
Open project Status.vbp              -- VB 4, 5, 6
Open project Status.vbproj           -- Visual Studio (VB.NET)
Open project Status(VS2008).vbproj   -- Visual Studio 2008
Open project Status(VS2010).vbproj   -- Visual Studio 2010
Open project Status(VS2012).vbproj   -- Visual Studio 2012
Open project Status(VS2013).vbproj   -- Visual Studio 2013
```

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 DOCS directory.

```
Open project ReadSSL.VBP             - VB 4, 5, 6
Open project ReadSSL.vbproj          - Visual Studio (VB.NET)
Open project ReadSSL(VS2008).vbproj  - Visual Studio 2008
Open project ReadSSL(VS2010).vbproj  - Visual Studio 2010
Open project ReadSSL(VS2012).vbproj  - Visual Studio 2012
Open project ReadSSL(VS2013).vbproj  - Visual Studio 2013
```

4.4 IMAP-Only Example Programs

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

All of the POP3 programs in the above section will also work with IMAP servers after defining the symbol

```
CONNECT_TO_IMAP_SERVER
```

in the program source code before compiling.

4.4.1 ImapFlagsT

The **ImapFlagsT** 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]
```

```
Open project ImapFlagsT.VBP              - VB 4, 5, 6
Open project ImapFlagsT.vbproj           - Visual Studio (VB.NET)
Open project ImapFlagsT (VS2008).vbproj  - Visual Studio 2008
Open project ImapFlagsT (VS2010).vbproj  - Visual Studio 2010
Open project ImapFlagsT (VS2012).vbproj  - Visual Studio 2012
Open project ImapFlagsT (VS2013).vbproj  - Visual Studio 2013
```

4.4.2 ImapSearch

The ImapSearch 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
```

```
Open project ImapSearch.VBP              - VB 4, 5, 6
Open project ImapSearch.vbproj           - Visual Studio (VB.NET)
Open project ImapSearch (VS2008).vbproj  - Visual Studio 2008
Open project ImapSearch (VS2010).vbproj  - Visual Studio 2010
Open project ImapSearch (VS2012).vbproj  - Visual Studio 2012
Open project ImapSearch (VS2013).vbproj  - Visual Studio 2013
```

4.4.3 ImapMBtest

The ImapMBtest example program tests IMAP functions seeImapConnect, seeImapListMB, seeImapDeleteMB, seeImapCreateMB, and seeImapSelectMB.

```
Open project ImapMBtest.vbp          -- VB 4, 5, 6
```

5 Revision History

The SMTP/POP3/IMAP Email Engine DLLs (SEE32.DLL and SEE64.DLL) are written in ANSI C. All language versions of SEE (C/C++, Delphi, Visual Basic, PowerBASIC, FoxPro, Delphi, Xbase++, COBOL, and FORTRAN) use the same identical DLLs.

Version 1.0: June 22, 1998.

- Initial release.

Version 2.0: September 28, 1998.

- A major update adding POP3 capability.
- Another SMTP example program (BCAST).

Version 2.1: November 28, 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, SEE_GET_CONNECT_STATUS,
- 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 12, 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.
- VB Class "seeClass" added.

Version 3.1: July 16, 1999.

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

Version 3.2: January 17, 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 and CODETEST examples.
- Added support for Lcc-Win32.
- Added seeCommand function.

Version 3.3: October 3, 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: July 17, 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 12, 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.

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: January 21, 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 3, 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 1, 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 ImapFlagsT, ImapSearch, and ImapMBtest example programs.
- Pass NULL for filename to seePop3Source / seeImapSource to revert back to server processing.

Version 5.1: May 6, 2009 (Win32 and Win64) versions

- 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)
- 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.

Version 5.2: March 9, 2010 (Win32 and Win64) versions

- 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.
- Added seeTestFileSet function to verify the existence of files.
- Added seeGetTicks function to return system ticks.
- Added seeSleep function (for languages not having a native Sleep call).

Version 6.0: February 8, 2011 (Win32 and Win64) versions

- 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 3, 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()
- Use large buffer (64K) for IMAP server response on channel 0

Version 7.1: April 4, 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: August 29, 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 2, 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: March 31, 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: September 15, 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: October 10, 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.