SMTP/POP3/IMAP Email Engine

Library for Visual dBase

Programmer's Manual

(SEE4DB)

Version 8.6

April 15, 2025

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 dBASE (SEE4DB)** library is a toolkit that allows software developers to quickly develop SMTP and POP3/IMAP email applications in Visual dBASE or dBASE Plus.

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 DLL library and 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.

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

The **SMTP/POP3/IMAP Email Engine for Visual dBASE** component library supports all Win32 (dBASE 7.0, 7.5 and dBASE Plus) versions of dBASE. **SEE4DB** includes numerous example programs that demonstrate SMTP and POP3/IMAP email functions used to create mail applications using the **SEE4DB** library.

SEE4DB runs under all versions of Windows through 32-bit and 64-bit Windows 10. The **SMTP/POP3/IMAP Email Engine SDK** DLLs (SEE32.DLL or SEE64.DLL) can also be used from any language (C/C++, .NET, Visual Basic, VB. NET, VBA, Delphi, Visual FoxPro, COBOL, PowerBASIC, Xbase++, etc.) capable of calling the Windows API.

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

- SEE4DB is a standard Windows DLL (NOT an OCX or ActiveX control) and is much smaller than a comparable OCX or ActiveX control.
- SEE4DB does NOT depend on ActiveX or similar "support" libraries.
- SEE is fully threadable.
- SEE4DB 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++ and .NET (SEE4C), Delphi (SEE4D), PowerBASIC (SEE4PB), Visual FoxPro (SEE4FP), Xbase++ (SEE4XB), COBOL (SEE4CB) and Visual Basic or VB.NET (SEE4VB). 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 environment.

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

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

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

1.1 Features

Some of the many features of **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 mail server, without reading the entire email.
- Delete any email on the POP/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
- Run up to 32 independent WIN32 threads concurrently.
- Can send email to mail addresses on a distribution list.
- Supports SMTP (ESMTP) and POP3 authentication.
- Set return receipt; add TO, CC, BCC recipients
- Set minimum and maximum wait times for server response.
- Supports ISO-8859 (European character sets) and UTF-8 (16 bit character sets) messages.
- Supports CHARSET_WIN_1250, CHARSET_WIN_1252 and CHARSET_WIN_1255.
- Can specify custom Content-Types; add custom header fields
- Use with GMAIL/Yahoo/Hotmail servers requiring SSL with (free) STUNNEL proxy server.
- Includes 70 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.
- Remove contents of attachments before writing to disk.
- 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.
- Supports all versions of 32-bit dBASE (V7.0 through V7.5) and dBASE Plus .
- Can be used with Microsoft Visual .NET and Visual C# (Managed Code)
- 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, Visual FoxPro, Delphi, Xbase++, Visual Basic, COBOL, Access and Excel.
- Is compatible with “Managed Code”.
- Works with Windows XP through Windows 11.
- License covers all programming languages.
- Royalty free distribution (no run-time fees) with your compiled application
- Evaluation versions are fully functional. No unlock code is required.
- Supports AUTH2 access delegation

Registration includes one year of free updates and technical support.

A good selection of dBase 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.
CODETEST	Base64 encodes/decodes strings.
FORWARD	Forwards undecoded email.
FROM	Displays header information for email on server.
GB2312	Sends email that is GB2312 (simplified Chinese).
GETRAW	Downloads specified email without decoding.
HELLO	Emails a short message.
HTML	Sends html encoded email with attachments.
ISO8859	Sends ISO-8859 encoded message and subject line
MAILER	Sends email with optional attachment.
MAILER2	Sends email with optional attachment using the "direct" method.
QUICK	Send email. Similar to Mailer except it uses a form.
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.
STATUS2	Lists all email on server. Similar to STATUS but uses the "direct method."
TESTCONN	Tests connection to a specified server and port.
ImapFlagT	Tests manipulation of flaps on IMAP server.
ImapSearch	Tests IMAP search capability.
MakeXOAuth2	Creates password from AUTH2 token.

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_4DB) in the set.

- SEE4DB Programmer's Manual (SEE_4DB.PDF)
- SEE User's Manual (SEE_USR.PDF)
- SEE Reference Manual (SEE_REF.PDF)

The SEE_4DB Programmer's Manual (SEE_4DB) is the language specific (Visual dBase) manual. All dBASE dependent programming issues such as compiling, compilers and example programs are discussed in this manual.

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

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

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

http://www.marshallsoft.com/see4db.htm

1.3 Example Program

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

```
#INCLUDE KEYCODE.CC
#INCLUDE SEE32.CC

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

? "HELLO"
Code = seeAttach(1, SEE_KEY_CODE)
if Code < 0
  ? "Cannot attach SEE"
  return
endif
Code = seeStringParam(0, SEE_LOG_FILE, DiagFile)
*** set maximum connect wait to 30 seconds
Code = seeIntegerParam(0, SEE_CONNECT_WAIT, 30000)
*** connect to POP3 server
? "Connecting to " + SmtpServer
Code = seeSmtpConnect(0, SmtpServer, SmtpFrom, SmtpReply)
if Code < 0
  Temp = SPACE(128)
  Code = seeErrorText(0,Code,Temp,128)
  ? Left(Temp,Code)
else
  *** send email message
  ? "Sending email to " + SmtpTo
  Code = seeSendEmail(0,SmtpTo,"","","Example Program","Hello from dBase","")
  if Code < 0
    Temp = SPACE(128)
    Code = seeErrorText(0,Code,Temp,128)
    ? Left(Temp,Code)
  else
    ? "Email has been sent."
  endif
endif
? "Done."
Code = seeClose(0)
Code = seeRelease()
return
```

In the example program above, **seeAttach** is called to initialize **SEE** and then **seeSmtpConnect** is called to connect to the SMTP mail host. **seeSendEmail** is then called, passing the addressee lists. The primary addressee is provided in the "To List". 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 SEE4DB, a dBASE compiler (any 32-bit version) should already be installed on your system and tested.

(2) Unzip SEE4DB86.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 SEE4DB files and copy SEE32.DLL to your Windows directory.

The SETUP installation program creates three sub-directories (default \SEE4DB) as follows:

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

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

1.5 Uninstalling

Uninstalling SEE4DB 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 SEE4DB.

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

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

After one year, 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 /SEE4DB/DOCS directory for more information.

2 Library Overview

The **SMTP/POP3/IMAP Email** component library has been tested on multiple computers running 32-bit and 64-bit Windows XP through Windows 10.

The SEE4DB library has been tested and works with all versions of 32-bit Visual dBASE and dBASE Plus.

2.1 Dynamic Link Libraries

The **SMTP/POP3/IMAP Email** component library uses 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

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 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 SEE4DB uses the same calling conventions and file naming conventions as the Win32 API. In particular, function names are NOT decorated. There are neither leading underscores nor trailing "@size" strings added to function names.

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

2.4 INCLUDE Files

All example programs include two files; `KEYCODE.CC` and `SEE32.CC`. The file `SEE32.CC` contains all the necessary constants and function declarations for SEE4DB, while the file `KEYCODE.CC` contains your key code.

There are three recommended ways to handle these INCLUDE files in dBase programs.

1. Copy the INCLUDE files to the dBASE compiler’s INCLUDE directory.
2. Edit the INCLUDE statements (for each program) with their physical location.
3. Replace the INCLUDE statements (in each program) by their contents.

2.5 Visual dBase 5.6 and 5.7

SEE4DB does not support Win16 applications.

2.6 Visual dBase 7.0 and above

Visual dBase 7.0 and above, including dBASE Plus, create 32-bit applications, and use the 32-bit DLL SEE32.DLL. Copy the 32-bit CC file, SEE32.CC to the dBASE compiler's INCLUDE directory.

2.7 Dynamic Strings

The Visual dBase language uses a technique known as "garbage collection" to manage string space at runtime, and may be called internally at any time by the dBase 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:

```
Code = seeSmtpConnect(0,SmtpServer,SmtpFrom,SmtpFrom)
if Code < 0
  * allocate buffer just before call
  Temp = SPACE(128)
  * put text in Temp
  Code = seeErrorText(1,Code,Temp,128)
  ? Left(Temp,Code)
endif
```

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.

2.8 Using Internal Memory

This section applies **ONLY** to using **DIRECT** mode as discussed in the section "Theory of Operation" in the SMTP/POP3/IMAP User's Manual, (`/docs/see_usr.pdf` or `http:/www.marshallsoft.com/see_usr.pdf`).

The dBASE dynamic string management functions (as discussed in Section 2.7 above) have another side effect when running in DIRECT mode (calling **seeDriver**). If Visual dBase moves memory at runtime, then memory references by dBase will use the new (moved) memory location, although SEE itself will still be using the original memory location previously passed to it. To work around this problem with Visual dBase (and other languages that do dynamic string management), **seeGetEmailLines** can be instructed to use its own memory:

```
Code = seeGetEmailLines(Chan, MessageNumber, 0, 0, max_buf_size)
```

If the fourth argument is 0, SEE will use its own memory. After **seeDriver** has been called to completion, the internal buffer can be copied by calling

```
Buffer = SPACE(max_buf_size)
Code = seeDebug(0, SEE_COPY_BUFFER, Buffer, max_buf_size)
```

seeGetEmailLines is the only function, which requires this technique, since there is no reason to use direct mode in other functions (such as **seeErrorText**) that use return buffers. See the program STATUS2.PRG for an example of using **seeGetEmailLines** in direct mode.

2.9 Error Display

Every SEE functions returns an integer value. Negative return values are always errors.

The error message text associated with **SEE** error codes can be found by calling **SeeErrorText**. Each sample program contains examples of error processing. For example,

```
*(assume ErrCode < 0 is returned by a SEE function)
Temp = SPACE(128)
* copy SEE error text to 'Temp'
Code = seeErrorText(0, ErrCode, Temp, 128)
* display the error
? Left(Temp, Code)
```

Also see 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 dBase** component library supports all versions of 32-bit dBASE (Version 7.0, Version 7.5 and dBASE Plus).

3.1 Compiling dBase Programs

dBASE programs end with the extension ".PRG". Before compiling any of the example programs, edit them with your email parameters. Programs can be edited within any text editor, and compiled from the dBASE command window with the COMPILE command (e.g.: COMPILE STATUS.PRG) or executed from the dBASE command window with the DO command (e.g.: DO STATUS.PRG).

To open a program within Visual dBase source editor, choose "File", then "Open". When the "Open File" dialog box appears, choose "Programs" for "Files of Type", then choose the program (*.PRG) to open. Lastly, choose "Open in Source Editor" for "Action" and push the "Open" button.

After editing the source program with your internet (or TCP/IP) parameters, you are ready to compile. From the dBase menu bar, choose "Build", then "Compile". To run choose, "Run". The dBASE command window must be displayed in order to view the output.

3.2 Compiling dBase Projects

Visual dBase projects consist of several types of files such as forms, reports, data modules, etc. The project file itself ends with the extension of ".PRJ".

There are two example dBase projects (QUICK and FROM). Open QUICK (or FROM) by choosing "File", then "Open Project" from the dBase menu bar. To compile QUICK, choose "Build" from the menu bar, then "Rebuild All". This will create QUICK.EXE, which can be executed by choosing "Execute quick.exe" from the "Build" menu bar pulldown, or from the Windows command line prompt.

3.3 Creating Executables

dBASE programs end in ".PRG". They can be compiled to an executable using the dBase BUILD command.

For example, to create SEEVER.EXE from SEEVER.PRG in the C:\SEE4DB\APPS directory, type the following in the dBase command window:

```
COMPILE C:\SEE4DB\APPS\SEEVER
BUILD C:\SEE4DB\APPS\SEEVER TO C:\SEE4DB\APPS\SEEVER
```

4 Example Programs

Each example program, with the exception of SEEVER.PRG, must be edited with your TCP/IP email parameters before compiling. Refer to the SMTP/POP3/IMAP User's Manual (SEE_USR) at (`/docs/see_usr.pdf` or `http:/www.marshallsoft.com/see_usr.pdf`) for details on email parameters.

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

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

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

4.1.2 CODETEST

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

4.1.3 TESTCONN

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

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

4.2 SMTP Example Programs

There are ten (10) SMTP email example programs. SMTP programs send email using an SMTP server.

4.21 FORWARD

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

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

4.2.2 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).

4.2.3 HELLO

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

Compare HELLO with the MAILER example program.

4.2.4 HTML

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

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

4.2.5 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).

4.2.6 MAILER

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

4.2.7 MAILER2

The MAILER2 example program operates the same as the MAILER program, except that it uses the "direct" method of calling seeDriver. MAILER2.PRG 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) or online at http:/www.marshallsoft.com/see_usr.pdf. MailSSL.PRG 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 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 a PDF file (*.pdf).

4.2.10 QUICK

QUICK is similar to MAILER, except that it accepts all necessary Internet TCP/IP settings using a form at runtime. The QUICK example program does not implement SMTP Authentication.

QUICK is organized as a dBASE project.

4.3 POP3/IMAP Example Programs

There are nine (9) POP3/IMAP example programs. These examples read email using a POP3 or IMAP server. Each example program, except FROM.PRG, must be edited with your email parameters before compiling.

4.3.1 FROM

FROM is similar to STATUS, except that it accepts all necessary Internet TCP/IP settings using a form at runtime. FROM is organized as a dBASE project.

4.3.2 GETRAW

GETRAW is an example program that downloads a specified email message without decoding it (in "raw" format). This is used to see what the email looks like on the server.

4.3.3 READER

READER can read email, including multiple MIME attachments, from your POP3/IMAP server, deleting each email after being read.

4.3.4 READER2

The READER2 example program operates the same as the READER program, except that it uses the "direct" method of calling seeDriver.

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 DOCS directory or online at http:/www.marshallsoft.com/see_usr.pdf. ReadSSL.PRG must be edited with your email parameters before compiling.

4.3.6 POP3RD

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

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

4.3.7 STATUS

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

4.3.9 STATUS2

The STATUS2 example program operates the same as the STATUS program, except that it uses the "direct" method of calling seeDriver.

4.4 IMAP-Only Example Programs

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

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]
```

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
```

5 Revision History

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

Version 3.0: May 10, 1999.

- Initial release of dBase version.

Version 3.1: August 5, 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.

Version 3.2: February 17, 2000.

- 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 AUTO examples.
- Added seeCommand function.
- Added internal memory usage support.

Version 3.3: November 29, 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 10, 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 4, 2002

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

Version 3.6: April 14, 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 17, 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: August 8, 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 22, 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 28, 2009 (Win32 and Win64 (x64)) 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) [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.
- Added Win64 DLL to support x64. [Visual C++ and Visual Basic version].

Version 5.2: March 27, 2010 (Win32 and Win64) versions

- 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 9, 2011

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

Version 7.0: November 15, 2011

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

Version 7.1: April 11, 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 18, 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 10, 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 18, 2016

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

Version 7.5: October 18, 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 24, 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: March 12 , 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 20, 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 13, 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: February 20, 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.
- Added Message-ID header to outgoing email.

Version 8.5: October 2, 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.
- Fixed crash attempting to open a non-existent file (64 bit only).
- Streamlined termination process when file could not be created or opened.

Version 8.6: April 15, 2025

- Increased password buffer to 3072 (3K), to support XOAUTH2 protocol.
- Added function seeMakeXOAuth2N()
- Added functions seeEncodeBase64() and seeDecodeBase64()
- Added diagnostics to log file for XOAUTH2 protocol
- Added errors codes SEE_AUTH_SHORT, SEE_AUTH_BAD_BEG, and SEE_AUTH_BAD_END
- Added functions seeShortToByte2() and seeByteToShort2()