Codestone Ltd

Internet Mail Client Control Library SSL Supplement

Welcome to the Internet Mail Client Control Library SSL Supplement we hope you will find the library to be a rapid and painless way to add SSL email functionality to your application.

CSMail/SSL adds Secure Socket Layer (SSL) and Transport Layer Security (TLS) functionality to the SMTPClient and POP3Client objects of Codestone's Internet Mail Client Control Library.

By utilising ActiveX COM technology the CSMail library is available to programmers working in Visual Basic ®, Visual C++ ®, Microsoft® Office, ASP on IIS, the Windows Scripting Host and many more environments. Our customers have used CSMail to great effect in applications including ISS Web-Mail sites, email gateways and bespoke email clients.

1 Introduction

CSMail/SSL adds Secure Socket Layer (SSL) and Transport Layer Security (TLS) functionality to the SMTPClient and POP3Client objects of Codestone's Internet Mail Client Control Library.

SSL and TLS provide protection for data passing between email clients and servers through strong symmetric encryption, validation of the server identity and optional validation of the client identity.

CSMail/SSL uses Microsoft's standard SSPI and Crypto-API frameworks to provide tight integration with the security and certificate management facilities of Microsoft Windows.

CSMail/SSL is designed to offer the developer a simple yet flexible model for managing SSL connections. In most cases SSL support can be added to application with only two or three lines of code.

New objects and features for CSMail/SSL

CSMail/SSL introduces several new features and objects to the library.

Advanced Handlers

CSMail/SSL introduces the **`SetAdvancedHandler`** method to the SMTPClient and POP3Client objects. Advanced handlers provide a mechanism for our developers to provide enhanced features, such as SSL/TLS, to the library while retaining maximum compatibility with applications using previous versions of the library.

Establishing SSL Connections – the SSLInfo Advanced Handler

The SSLInfo advanced handler object encapsulates the parameters and attributes of an SSL/TLS connection. Developers can specify an SSL/TLS connection by associating an SSLInfo object with an SMTPClient or POPClient object (through the **`SetAdvancedHandler`** method).

See the SSLInfo Object Developers Reference below for more information.

Library diagnostic logging - LogHandler Advanced Handler

The LogHandler advanced handler object provides an event-based mechanism for the developer to receive logging and diagnostic messages from the library. Developers can associate a LogHandler object with an SMTPClient or POPClient object (through the **`SetAdvancedHandler`** method).

See the LogHandler Object Developers Reference below for more information.

Detailed Error Information - ErrorInfo Object

As existing developers will be aware, the library uses the standard VB error handling mechanism to notify applications of network, sever and other errors. The original range of errors defined by the library provides a broad indication of the cause of the error and is sufficient for most applications in most cases. In some cases, however, it is desirable to have a more detailed indication of the cause of the error.

The **`ExtendedError`** property of the SMTPClient and POPClient objects provides access to more detailed error information in these cases.

See the ErrorInfo Object Developers Reference below for more information.

Usage

Enabling SSL/TLS

CSMail/SSL is designed to be a simple as possible for the developer to incorporate into an application. In most cases you will need only to instantiate an SSLInfo object and associate it with an SMTPClient or POP3Client object with the **SetAdvancedHandler** method.

```
Dim smtp as new CSMailLib.SMTPClient
Dim ssl As New CSMailLib.SSLInfo
...

' Use SSL for this connection
smtp.SetAdvancedHandler ssl

smtp.Connect
...
```

Setting parameters for a secure connection

When you instantiate an SSLInfo object it is initialised with a set of default values that make sense for the vast majority of connections. See the **Options()** property array in the SSLInfo Object Developers reference for information on the parameters you can set for the connection.

Getting information about an established SSL connection

After a successful SSL/TLS connection has been established to a server you can retrieve information about the negotiated security parameters from the SSLInfo object. There is no requirement that you fetch any of this information but you may find it useful for diagnostic or information purposes. See the **Options()** property array in the SSLInfo Object Developers reference for information on the attributes you can fetch.

Diagnosing SSL Errors

See Appendix A. Common SSL/TLS Error conditions and their causes.

SSLInfo Object Developer's reference

Mode Property

Declaration: Mode as SSL_MODE

Description

Specifies if the SMTPClient or POP3Client object will use a secure channel or the protocol extensions to establish the secure connection.

Values

MODE_CHANNEL	Establish a secure channel. **Default**.
MODE_STARTTLS	Use the protocol extensions to secure a connection.

Example

```
Dim ssl As New CSMailLib.SSLInfo
Dim smtp as new CSMailLib.SMTPClient

...

' Use STARTTLS protocol extension rather than a secure channel
ssl.Mode = MODE_STARTTLS

...

smtp.SetAdvancedHandler ssl
```

Option Property Array

Declaration: Option(Index as Long) as VARIANT

Description

The **`Option`** property array provides access to a number of advanced parameters and attributes of a secure connection. In most cases you will not need to use the **`Option`** property array in your application.

Before establishing a secure connection you can set some of the options to specify parameters to be used for the connection. If you don't specify a parameter the library will use a sensible default and you should exercise great caution before changing any parameter from its default; doing so may reduce the level of security for the connection.

After a secure connection has been successfully established you can determine various attributes of the connection through some of the options.

Property Index Values

Input parameters	
SSLNoServerNameCheck	Prevents the library from comparing the supplied target name with the subject names in server certificates. Use this option, with caution, when the hostname specified in the call to **`Connect()`** does not match the principal name on the server certificate. The default value for this option is False.

SSLNoServerCertificateValidation	Prevents the library from validating the received server certificate chain. Use this option, with extreme caution, if you need to connect to a server that does not have a verifiable security certificate. The default value for this option is false. The default value for this option is False.
SSLRevocationCheckAll	When validating a certificate chain the library will check all certificates for revocation. The default value for this option is True.
SSLRevocationCheckEnd	When validating a certificate chain the library will check only the last certificate for revocation. The default value for this option is False.
SSLRevocationExcludeRoot	When validating a certificate chain the library will not check the root for revocation. The default value for this option is False.
SSLRevocationIgnoreNRC	When checking for revoked certificates the library will not report errors when it is not possible to do a revocation check on the certificate. The default value for this option is True.
SSLRevocationIgnoreRO	When checking for revoked certificates the library will not report errors when It is not possible to connect to the revocation server. The default value for this option is True.
SSLMinCipherStrength	Set the minimum symmetric encryption cipher strength for subsequent connections. If this value is zero the library will choose an appropriate cipher strength. The default for this option is zero; that is the library will choose an appropriate value.
SSLMaxCipherStrength	Set the maximum symmetric encryption cipher strength for subsequent connections. If this value is zero the library will choose an appropriate cipher strength. The default for this option is both zero; that is the library will choose an appropriate value.

<table>
<tr><td>SSLConnectionProtocolIn</td><td>An array of the protocols supported for subsequent connections. If the array is empty the library selects an appropriate protocol. Valid values for the elements of the array are:
<table><tr><td>SSL_TLS1</td><td>Transport Layer Security version 1.0</td></tr><tr><td>SSL_SSL3</td><td>Secure Sockets Layer version 3.0</td></tr><tr><td>SSL_SSL2</td><td>Secure Sockets Layer version 2.0</td></tr><tr><td>SSL_PCT1</td><td>Private Communications Technology version 1.0</td></tr></table>
The default for this option is an empty array; that is the library will choose an appropriate protocol.</td></tr>
<tr><td>SSLConnectionCipherIn</td><td>An array of the symmetric ciphers supported for subsequent connections. If the array is empty the library selects an appropriate cipher. Valid values for the elements of the array are:
<table><tr><td>SSL_RC2</td><td>RC2 block encryption algorithm</td></tr><tr><td>SSL_RC4</td><td>RC4 stream encryption algorithm</td></tr><tr><td>SSL_DES</td><td>DES encryption algorithm</td></tr></table>
The default for this option is an empty array; that is the library will choose an appropriate cipher.</td></tr>
<tr><td>SSLConfidentiality</td><td></td></tr>
<tr><td>SSLClientCertificateName</td><td></td></tr>
</table>

<table>
<tr><td colspan="2">Output attributes</td></tr>
<tr><td>SSLConnectionProtocolOut</td><td>The protocol used to establish the connection. One of the following values:
<table><tr><td>SSL_TLS1</td><td>Transport Layer Security version 1.0</td></tr><tr><td>SSL_SSL3</td><td>Secure Sockets Layer version 3.0</td></tr><tr><td>SSL_SSL2</td><td>Secure Sockets Layer version 2.0</td></tr><tr><td>SSL_PCT1</td><td>Private Communications Technology version 1.0</td></tr></table></td></tr>
</table>

<table>
<tr><td>SSLConnectionCipherOut</td><td>The symmetric encryption algorithm used to protect the data during the connection. One of the following values:<table><tr><td>SSL_RC2</td><td>RC2 block encryption algorithm</td></tr><tr><td>SSL_RC4</td><td>RC4 stream encryption algorithm</td></tr><tr><td>SSL_DES</td><td>DES encryption algorithm</td></tr><tr><td>SSL_SKIPJACK</td><td>Skipjack block encryption algorithm</td></tr></table></td></tr>
<tr><td>SSLCipherStrength</td><td>The strength of the symmetric encryption cipher; 0, 40, 56, 80, 128, or 168 bits.</td></tr>
<tr><td>SSLHash</td><td>The hash used for generating message authentication codes (MACs). One of the following values:<table><tr><td>SSL_MD5</td><td>MD5 hashing algorithm</td></tr><tr><td>SSL_SHA</td><td>SHA hashing algorithm</td></tr></table></td></tr>
<tr><td>SSLHashStrength</td><td>The strength of the hash; 128 or 160 bits.</td></tr>
<tr><td>SSLExchange</td><td>The key exchange algorithm used to generate the shared master secret for the symmetric encryption. One of the following values:<table><tr><td>SSL_RSA</td><td>RSA key exchange.</td></tr><tr><td>SSL_DH</td><td>Diffie-Hellman key exchange</td></tr></table></td></tr>
<tr><td>SSLExchangeStrength</td><td>The strength of the exchange mechanism.</td></tr>
</table>

Examples

```
' Example 1
' Stop the client from verifying the server' name
' with that on its certicate.
' (Do NOT use this option in production code!!!!)
'
ssl.Option(SSLNoServerNameCheck)=True


' Example 2
' Restrict the choice of protocols to SSL1 & SSL2
'
Dim protocols(2) As Long
...
protocols(0)=SSL_SSL2
protocols(1)=SSL_SSL3

ssl.Option(SSLConnectionProtocolIn)=protocols
```

ErrorInfo Object Developer's reference

Summary

The ErrorInfo object allows the developer to retrieve a greater degree of detail about the VB errors raised by the SMTPClient and POP3Client objects.

This object has been introduced to allow developers to better determine the cause of an error condition that the library has raised through the normal VB error handling mechanism. CSMail/SSL also uses this functionality to provide more descriptive errors than was previously the case.

The extended error information is available through the ExtendedError property of the SMTPClient and POP3Client objects.

Number Property

Declaration: Number as Long

Description

The extended error number, the following tables lists the possible values for this property.

VB constant	Hex	Dec	Notes
eexProxyNoconn	0x03e9	01001	The object was unable to create a TCP connection to the proxy server
eexProxySocketerror	0x03ea	01002	A socket error occurred while communicating with the proxy server
eexProxyServerclosedport	0x03eb	01003	The proxy server unexpectedly closed the connection
eexProxyException	0x03ec	01004	An internal error occurred while communicating with the proxy server
eexProxyUnexpecteddefault	0x03ed	01005	An unexpected error occurred while communicating with the proxy server
eexProxyRequestfailed	0x03ee	01006	The proxy server rejected the request
eexProxyAuthrequired	0x03ef	01007	The proxy server requires authentication but no credentials have been supplied
eexProxyAuthtypeunknown	0x03f0	01008	The proxy server does not support the authentication mechanism requested by the client
eexProxyAuthfailed	0x03f1	01009	The proxy server has rejected the clients credentials.
eexProxyAuthnologon	0x03f2	01010	
eexNoExtended	0x044d	01101	
eexDNS	0x044e	01102	The server or proxy name cannot be found in the DNS
eexTCPConnect	0x044f	01103	The client cannot connect to the server or proxy
eexPopServer	0x04b1	01201	The POP3 server has reported an error
eexSocketTx	0x0450	01104	A network error has occurred while sending data

eexSocketRx	0x0451	01105	A network error has occurred while receiving data
eexSocket	0x0452	01106	A network error has occurred
eexNotConnected	0x0453	01107	The object is not connected to the server or proxy
eexSmtpServer	0x0515	01301	The SMTP server has reported an error
eexSSLW32	0x0579	01401	An error has occurred during an SSL operation
eexSSLConnected	0x057b	01403	Attempt to change the properties of an existing SSL connection
eexSSLNocontext	0x057c	01404	Attempt to perform an operation on an un-established SSL connection
eexSSLCertName	0x057d	01405	The name on the host's certificate does not match its hostname
eexSSLProtocol	0x057e	01406	An SSL Protocol Error occurred
eexSSLCertificate	0x057f	01407	The certificate is invalid

Text Property

Declaration: Text as String

Description

A textual description of the extended error.

OSError Property

Declaration: OSError as Long

Description

The native windows error code.

LogHandler Object Developer's reference

Summary

The LogHandler object provides a mechanism for developers to integrate diagnostic information from CSMail/SSL into their own diagnostic code.

Level Property

Declaration: Level as Long

Description

Specifies the maximum level of detail which the LogHandler object will provide:

LOG_S_FATAL	The LogHandler object provides messages only relating to fatal incidents
LOG_S_WARNING	The LogHandler object provides messages relating to warning conditions
LOG_S_INFORMATION	The LogHandler object provides detailed informational messages

LOG_S_INFORMATION provides the greatest detail.

Facility Property

Declaration: FacilityFlags as Long

Description

Specifies which facilities the library will generate messages about.

LOG_F_PROTOCOL	SMTP/POP3 protocol
LOG_F_SSL	SSL connections
LOG_F_PROXY	Proxy connections
LOG_F_USER	User messages

OnLogMessage Event

Declaration: OnLogMessage (*Severity* as Long, *Facility* as Long, *Message* as String)

Parameters

Severity	*The severity level of the Message, see the Level property.*
Facility	*The facility which generated the message, see the Facility property*
Message	*The log Message*

Description

This method is called with logging messages from the object.

Additional methods and properties for the SMTPClient Object

SetAdvancedHandler Method

Declaration: SetAdvancedHandler(*AdvancedHandler* as Object)

Parameters

AdvancedHandler	*An object. Currently supported types are:*
	An SSLInfo object
	A LogHandler object
	A SOCK4ProxyInfo object
	A SOCK5ProxyInfo object
	A POP3ProxyInfo object
	An SMTPProxyInfo object

Description

Adds an advanced handler to the SMTPClient object.

Return Value

There is no return value from this method. If any error occurs while saving the message an error will be raised and should be handled through the VB/VBA/VBScript On Error mechanism.

Example

ExtendedError Property

Declaration: ExtendedError as ErrorInfo

Description

Provides access to the ExtendedError object for the SMTPClient object. See the ErrorInfo Object Developers reference for more information on the extended error information.

Additional methods and properties for the POP3Client Object

SetAdvancedHandler Method

Declaration: SetAdvancedHandler(*AdvancedHandler* as Object)

Parameters

AdvancedHandler	*An object. Currently supported types are:*
	An SSLInfo object
	A LogHandler object
	A SOCK4ProxyInfo object
	A SOCK5ProxyInfo object
	A POP3ProxyInfo object
	An SMTPProxyInfo object

Description

Adds an advanced handler to the POP3Client object.

Return Value

There is no return value from this method. If any error occurs while saving the message an error will be raised and should be handled through the VB/VBA/VBScript On Error mechanism.

ExtendedError Property

Declaration: ExtendedError as ErrorInfo

Description

Provides access to the ExtendedError object for the POP3Client object. See the ErrorInfo Object Developers reference for more information on the extended error information.

Appendix A. Common SSL/TLS Error conditions and their causes

<table>
<tr><th rowspan="2">VB Error Code</th><th colspan="2">ExtendedError</th><th rowspan="2">Notes</th></tr>
<tr><th>.Number</th><th>.OSError</th></tr>
<tr><td>errConnect</td><td>eexDNS</td><td>Any</td><td>The server name cannot be resolved.
Check the server name.</td></tr>
<tr><td>errConnect</td><td>eexTCPConnect</td><td>Any</td><td>There is no response from the server on the given port.
Check that the port is correct for the server you are trying to connect to. The default ports for SMTP/POP3 servers are:
<table>
<tr><th>Protocol</th><th>Type</th><th>Port</th><th>Protocol</th><th>Type</th><th>Port</th></tr>
<tr><td>SMTP</td><td>None</td><td>25</td><td>POP3</td><td>None</td><td>110</td></tr>
<tr><td>SMTP</td><td>SSL Channel</td><td>465</td><td>POP3</td><td>SSL Channel</td><td>995</td></tr>
<tr><td>SMTP</td><td>STARTTLS</td><td>25</td><td>POP3</td><td>STLS</td><td>110</td></tr>
</table></td></tr>
<tr><td>errConnect</td><td>eexSSLProtocol</td><td>Any</td><td>You may have tried to establish a secure channel connection on a server that does not support secure channels.
Check that the server supports secure channel connections and which port you should be connecting on.</td></tr>
<tr><td>errConnect</td><td>eexSSLCertName</td><td>Any</td><td>The name supplied to the call to Connect() does not match the name on the server’s certificate.
Check that you are specifying a name rather than an IP address for the server.
Check that the server’s certificate is correct.
In debugging contexts you can use the SSLNoServerNameCheck option to temporarily disable this error. You should never use SSLNoServerNameCheck in production code.</td></tr>
</table>

errConnect	eexSSLCertificate	Any	The server's certificate is invalid. Check that the server's certificate is correct. In debugging contexts you can use the SSLNoServerCertificateValidation option to temporarily disable this error. You should never use SSLNoServerCertificateValidation in production code.

Appendix B. An SSL primer

SSL/TLS in the context of SMTP and POP3 applications

It is important to note that in the context of email standards SSL does not offer endpoint to endpoint security: that is to say that only the connection between client and server applications are protected. Email will commonly be transported across the Internet through one or more relay servers and these relays will make their own decisions about secure transport. It would be fair to say that very few relay servers will use SSL/TLS while relaying mail. Nevertheless SSL is a useful mechanism for email applications and provides a high level of security for local systems. Some examples of situations in which SSL/TLS provides useful security are included in subsequent sections.

Features of a secure connection

As noted above SSL and TLS provide protection for session information and data passing between client and server applications. This protection is composed of the following elements.

Verification of server identity

It is vital for secure communications that the client application be able to verify the identity of the server: without this basic prerequisite there is little point in encrypting the session – an attacker could set up a system to masquerade as the true server and, since the encryption will be agreed between the client application and the server the attacker will be able to read the unencrypted data.

Encryption of Session Information and Data

It was noted above that, while SSL/TLS does not in itself provide endpoint-to-endpoint security for Internet email, it does provide important security features for client-server communication. Here are two increasingly common examples of situations where SSL/TLS provides useful security.

Wireless networks: many wireless networks are inherently insecure and usernames and passwords for SMTP and POP3 servers may be passed over the network with weak or non-existent encryption providing an opportunity for attackers to intercept these credentials and other sensitive data.

Mobile workers/Home workers: road warriors and home workers may access their corporate networks through the Internet presenting an opportunity for an attacker to intercept unencrypted credentials or sensitive data.

Verification of client identity

By requiring that server verifies the identity of the client system administrators can ensure that only clients which are verifiable beyond a simple username/password scheme can access a server.

An SMTP server can be configured to only accept mail from a set of clients, preventing attackers from subverting the SMTP server to send unauthorised messages.

A POP3 server can offer a high level of protection for access to a user's mailbox.

Public Key Certificates

Server and client identities are verified through public key certificates. Public key certificates contain information about their owner (such as name, location, organisation etc) and are signed (with public key signature technology) by a certificate authority. The issuing certificate authority may be a ***trusted*** certificate authority or it may itself have a certificate from another certificate authority. In such a way a chain of certificates is constructed, leading finally to a trusted certificate authority.

CSMail/SSL uses the standard Windows certificate management technology.

Certificate Revocation

A certificate issuer can revoke a certificate if the certificate should no longer be considered valid. It is not, however, always possible to definitively determine the revocation status of the certificate; here are some reasons why revocation status may be unobtainable.

The certificate authority may not provide revocation status information.

The certificate authority may provide revocation status information by periodically issuing a revocation list and the list may not be installed locally, may be out of date or may be temporarily unavailable.

The certificate authority may offer online revocation status information but the service may be temporarily unavailable.

CSMail/SSL will, by default, attempt to check the revocation status of a certificate and will report an error if the certificate can be determined to be revoked. It will, not, however report an error in cases where the revocation status of the certificate is unknown or temporarily unavailable. See the SSLRevocationIgnoreNRC and SSLRevocationIgnoreRO options in the SSLInfo Object Developers reference for more information.

Secure Connection Types

Secure connections with SMTP and POP3 servers may be established though one of two mechanisms, a secure channel or through extensions to the SMTP and POP3 protocols. The administrator of the SMTP or POP3 server will have made the choice of supported connection types.

CSMail/SSL supports both secure channel and the STARTTLS and STLS protocol extensions.

Secure channel communications

In the case of secure channel communications the entire session is protected by SSL/TLS. A secure service will typically listen on a different port to the standard, unprotected, service. In the case of Secure SMTP the server usually listens on port 465 rather than 25 and in the case of Secure POP3 the server will usual listen on port 995 rather than 110.

When establishing a secure channel session the client will establish a TCP connection to the secure service and immediately start the SSL/TLS negotiation to secure the session.

CSMail/SSL attempts to establish a secure channel by default, so you need to do nothing special in this case.

Protocol extensions: STARTTLS and STLS

In this case the client establishes a normal, unsecured, connection with the server and then issues a command to the server requesting that the connection be secured. The SMTP STARTTLS and POP3 STLS extensions are used to request this transition to a secure connection.

In most cases services supporting these protocol extensions will listen on the normal ports for the service (i.e. port 25 for SMTP and port 110 for POP3).

As the names of these protocol extensions suggest they will use only TLS protection.

See the Mode property in the SSLInfo Object Developers reference for information on how to use the STARTTLS and STLS protocol extensions.

The SSL/TLS handshake

When establishing a secure connection the client and server conduct a conversation during which they agree upon the protocol (SSL or TLS), the key exchange mechanism (RSA, Diffie-Hellman etc) and the symmetric encryption algorithm (RC2, RC4, DES etc) that will be used to secure the connection.

The details of the negotiation between the client and server vary according to the protocols, handshake and encryption used for the connection. All connections, however, perform the same basic operations.

The client starts the handshake by listing the protocols, exchange mechanisms and symmetric algorithms that it supports. The server makes an appropriate selection from the available options and sends this back to the client. The server will typically send its certificate with this data and the client verifies the authenticity of the certificate before continuing.

The client and server now either i) exchange a set of symmetric encryption keys or ii) exchange a shared secret and independently generate the symmetric keys from the secret. The keys or shared secret are protected with asymmetric (public key) encryption, typically the client will use the public key in the server's certificate to encrypt data it sends to the server and the server uses the certificate's private key for data it sends to the client.

If the server requires a valid certificate from the client this will also be exchanged and verified.

Once both the client and server have the symmetric encryption keys the handshake is completed and both client and server start encrypting all subsequent data with the symmetric encryption algorithm.

You can change the protocols etc that CSMail/SSL will advertise to the server through the .Option property of the SSLInfo Object. You can also discover the agreed parameters from the .Option property.

Further reading

Developers interested in learning more about SSL and TLS may be interested in **SSL and TLS Essentials: Securing the Web** by Stephen Thomas; published by John Wiley & Sons Inc; ISBN: 0471383546 (March 2000).

Although this book focuses on the web applications of SSL/TLS the early chapters include a clear introduction to the operation and features of SSL, Asymmetric and Symmetric encryption and Certificate management. An appendix provides a useful SSL Security Checklist. The middle section of the book focuses on the SSL and TLS protocols themselves in a clear though not overly technical fashion.