CSMail 1.5.5 Release Notes. 28 August 2002.

1 New Features

1.1 BodyBinary property (Section Object)

To date a shortcoming of the CSMail library has been the lack of a mechanism for simply accessing the binary contents of a message section. The `Body` property of the section object is unsuitable for accessing the binary content because Unicode transformations are carried out on the content when the property is accessed. This has been a particular inconvenience for ASP developers who may, for example, wish to send an attached image to the end-user's browser without the intermediate step of saving the section contents to a disk file.

The `BodyBinary` property allows direct access to the binary contents of a section without any Unicode translation.

1.1.1 BodyBinary Property

Declaration: BodyBinary as Variant

Compatibility: V 1.5.5 and greater

Description

The binary contents of the section body.

This read/write property allows access to the binary data held in the body of the section - this property always reflects the native (un-encoded) form of the data - the data is automatically encoded when the SMTP client sends it and decoded when the POP3 Client receives it.

The `BodyBinary` property can be used to safely access binary data without the undesirable transformations which conversion to and from Unicode can produce.

1.1.1.1.1 Example (ASP)

```
Response.ContentType = section.Header("Content-Type")
Response.addHeader "content-disposition", "filename=" & section.filename
response.binarywrite (section.bodybinary)
```

1.2 Corrected error codes in type library

A number of the values in the Error enumeration contained within the CSMail type library were incorrect. These values have now been corrected in V1.5.5.

1.3 Corrected required permissions for Message.LoadFromFile()

The `Message.LoadFromFile()` method could, under some circumstances, fail on read-only media – this has been corrected in V1.5.5.

1.4 Quick connect option for POP3Client

The default behaviour of the `POP3Client.Connect()` method is to establish a TCP connection to the server, issue the USER and PASS commands of the POP3 protocol and then issue STAT and LIST commands. In most circumstances this sequence of commands is appropriate and desirable.

There are, however, circumstances under where an application needs to only connect to a POP3 server and not to retrieve or carry out any other operations on the messages on the server. If this is the case - and the maildrop contains a very large number of messages - issuing these commands (particularly the LIST command) can impose an unnecessary overhead on the application. Two examples of these circumstances are POP3 before SMTP authentication and validating POP3 credentials.

In V1.5.5 a new option has been added to the Settings Object to prevent the `POP3Client` object from calling STAT and LIST during the a call to `Connect()`.

If the `optionQuickPop3Connect` option is enabled the STAT and LIST commands are not issued during a call to the `Connect()` method. Only a subset of the methods of the `POP3Client` object is subsequently available until the `POP3STAT()` method is called explicitly. The table below lists the properties and methods of the object and their behaviour when the `optionQuickPop3Connect` option is enabled.

Method/Property	**Notes**
`Connect()`	Inappropriate after `Connect()` and before `Close()`
`LastServerResponseString`	Functional
`Close()`	Functional
`POP3STAT()`	Functional
`ByteCount`	`errPOP3QuickConnectStat` (0x80040227) until `POP3STAT()` is called and un-initialised until `POP3LIST()` is called.
`MessageCount`	`errPOP3QuickConnectStat` (0x80040227) until `POP3STAT()` is called.
`POP3LIST()`	`errPOP3QuickConnectStat` (0x80040227) until `POP3STAT()` is called.
`POP3UIDL()`	`errPOP3QuickConnectStat` (0x80040227) until `POP3STAT()` is called.
`POP3RETR()`	`errPOP3QuickConnectStat` (0x80040227) until `POP3STAT()` is called.
`POP3DELE()`	`errPOP3QuickConnectStat` (0x80040227) until `POP3STAT()` is called.
`POP3NOOP()`	Functional
`POP3RSET()`	`errPOP3QuickConnectStat` (0x80040227) until `POP3STAT()` is called.
`POP3QUIT()`	Functional
`ConnectA()`	Inappropriate after `Connect()` and before `Close()`
`Messages`	`errPOP3QuickConnectStat` (0x80040227) until `POP3STAT()` is called.
`POP3TOP()`	`errPOP3QuickConnectStat` (0x80040227) until `POP3STAT()` is called.
`RetrieveMessages()`	`errPOP3QuickConnectStat` (0x80040227) until `POP3STAT()` is called.
`LastSocketError`	Functional
`SetProxyInfo()`	Inappropriate after `Connect()` and before `Close()`

Examples

```
' vb script asp etc
Set MailSettings=CreateObject("CSMail.Settings")
MailSettings(15)=true

' vb
Dim MailSettings As New CSMAILLib.Settings
MailSettings(optionQuickPop3Connect) = True
```

2 Technical Addendum to CSMail 1.5.5 Release Notes

As in previous releases, notably V 1.2, the addition of new properties to existing coclasses in V 1.5.5 required the definitions of new interfaces (so as to honour the immutability of COM interfaces). The new interfaces have, as previously, been defined so as to maintain binary compatibility with the existing interfaces.

Late bound (IDispatch) non-compiled environments (e.g. VBA, vbscript and ASP)

In late bound and non-compiled environments (for example VBA, vbscript and ASP) these new interfaces will clearly have no impact on existing code – beyond the positive effect of exposing new properties. Developers using the new properties exposed by the updated coclasses must, of course, be sure to distribute the new csmail.dll with their applications.

Early bound compiled environments (e.g. VB and C++)

In early bound and compiled environments (for example VB and C++) there will also be no negative impact on existing code.

- Existing early bound compiled code will continue to obtain the old interfaces on the updated coclasses.
- Early bound code compiled against V1.5.5 or greater will obtain the new interfaces on the updated coclasses. Developers using the new properties exposed by the updated coclasses must, of course, be sure to distribute the new csmail.dll with their applications.

Visual Basic developers should need to take no action to take advantage of the new properties exposed by the updated coclasses. It may, however, be necessary to remove the reference to 'Codestone Internet Mail Controls Type Library' (under Projects|References) and then reinsert the reference before recompiling the application.

C++ developers who do not require the functionality offered by the new interfaces can leave their application code unchanged – the existing interfaces are unchanged and still fully supported by the objects. C++ developers who wish to use the new functionality should be sure to `QueryInterface` for the new interfaces. The table below outlines the new interfaces on the effected objects.

Summary of new interfaces in V1.5.5

CoClass	**Interfaces supported prior to V1.5 (default interface in bold)**	**Interfaces supported in V1.5 (default interface in bold)**
CSMail.Section	**ISection**	ISection **ISection2**
CSMail.SectionCollection	**ISectionCollection**	ISectionCollection **ISectionCollection2**
CSMail.Message	**IMessage**	IMessage **IMessage2**
CSMail.MessageCollection	**IMessageCollection**	IMessageCollection **IMessageCollection2**
CSMail.POP3Client	IPop3Client **IPop3Client2**	IPop3Client IPop3Client2 **IPop3Client3**