SMTPClient Class for RealBasic / RealStudio

Thanks for your interest in the SMTPClient Class for RealBasic / RealStudio. This class allows you to send email messages to a smtp server, either through a normal connection or a secure ssl connection. The main difference with the class RealBasic/RealStudio has build in is that the SMTPClient class doesn't touch your email message.
You can construct the email messages the way you want: by using the EmailMessage Class of RealBasic/RealStudio, you're own method, or by importing an email you've created with Mail.app. It's all up to you.

SMTPEmailMessage
The SMTPClient Class uses it's own SMTPEmailMessage Class. It's an extended version of the EmailMessage Class of RealBasic/RealStudio. Besides all functionality of the standard EmailMessage Class it offers three additional properties:

- **MsgID As String**
  This is an identifier for your benefit. The SMTPClient Class doesn't use it itself, but when it's done sending a message it returns this MsgID, so you can keep track of the messages already sent.
- **MsgSource As String**
  This is the most important property of the SMTPEmailMessage Class. If not empty it will overwrite all properties of the standard EmailMessage Class. The MsgSource should contain the raw source a complete email message, including all headers.
- **RecipientList() As String**
  Although accessible from 'the outside' it's purpose is for internal use of the SMTPClient Class.

So, in general: you can use the SMTPEmailMessage Class as a substitute for the standard EmailMessage Class, but the moment you define the content of the MsgSource property this MsgSource property is the only one used.

SMTPClient
To use the SMTPClient Class, just add the SMTPClient folder to your project.
Next drag the SMTPClient object to the a window of your project.

In order to send the email message(s), the SMTPClient Class needs information about the smtp server it should use. For this, you need to set the following properties:

- **Address As String**
  The ip address or hostname of the smtp server
- **Port As Integer**
  The port number to use. In general it's 25 for normal connections and 465 or 587 for secure connections.
- **Username, Password As String**
  The user name and password needed to authenticate yourself on the smtp server. If you leave those empty no authentication will be used. If you specify them authentication will be used.
- **Secure As Boolean**
  Set to True if you want a secure connection, False for a plain connection. Note that you also need to set the right port number and connection type.
- **ConnectionType As Integer**
  Specifies the type of secure connection.
  ConnectionType can take the following values:

0 - SSLv2: SSL (Secure Sockets Layer) version 2.
1 - SSLv23: SSL version 3, but can roll back to 2 if needed.
2 - SSLv3: SSL version 3.
3 - TLSv1: TLS (Transport Layer Security) version 1.
The default value is 1, SSL version 23.

After setting all the server properties, you're ready to send email messages. There are two ways to do so.

1) To send a single message, construct an email using the SMTPEmailMessage Class and use the method **SendMessage(EmlMsg As SMTPEmailMessage)** to send it.
2) To send a multiple messages, construct those emails using the SMTPEmailMessage Class and use the method **AddMessage(EmlMsg As SMTPEmailMessage)** to add them to the queue. You can use the method **MsgCount() As Integer** to find out how many messages are in the queue. After adding all messages to the queue you can call the method **SendMessages()** to send them.

Events

The SMTPClient Class has the following events:

- **CommandSent(Command As String)**
  This event is fired each time a smtp-command is send to the server. Mostly for debugging purposes.
- **ReplyReceived(ReplyCode As String)**
  Same for replies from the smtp server.
- **ReplyError(ErrorCode As String)**
  Is fired when the smtp server replies with an error code.
- **Completed(Successful As Boolean)**
  This even is fired after the connection to the server is closed. The boolean SuccessFul wil let you know if everything went right (or wrong).
- **Connected()**
  Fired when a connection to the smtp server has been established.
- **Error()**
  Fired in case of errors
- **MailboxError(EmailAddress As String, ErrorCode As String)**
  This event is fired when the smtp server rejects an email address.
- **MessageProgress(BytesSent as Integer, BytesLeft As Integer, BytesTotal As Integer)**
  To prevent overflowing the tcp buffers email messages are send to the smtp server in chunks. The standard SendProgress() event will only give you information about a single chunk. The MessageProgress() event will give you progress information about the complete message being send.
- **MessageSend(MsgID As String)**
  Is fired after one message has been send.
- **MessagesSend()**
  Is fired after all messages have been send.

To make live easy for you, we've included three samples which will show you the different ways you can use the SMTPClient Class.

- Demo 1
  Simple demo. It uses the EmailMessage Class of RealBasic/RealStudio to create the message.
- Demo 2
  Displays a way to send raw messages. Create a message using your mail program. Drag the message from your mail program to your desktop.
  Next, drag the message from your desktop to the dropbox area of the sample application.
- Demo 3
  Demonstrates how you use canned messages.
  Two messages have been created: one with Microsoft Entourage and the other with Mail.app. They are dragged out of those applications and loaded into an editor. We've replaced the From and To email addresses with special keywords.
  Those modified email messages are dragged into the project.
  The demo application will fill in the special keywords before sending the email.

If you like the SMTPClient Class and want to use it in your application, you need to register it. It's only $ 23. Just send your browser to www.mrose.nl/smtpclient.

If you need help, have feature requests or know a way how we can improve it, we would like to hear from you. But, please use our support forum for it.

Regards,

Bert Rozenberg
Mountain Rose Multi Media