How to HTML and SMTP

by Randy Charles Morin

I was asked last month "I would like to know if there is an easy way to add attachments to the sent (SMTP) message?" Well there is no easy way, but this month we'll struggle thru this together. In my next two articles, I will discuss how to send html and various attachments using the set of socket classes I developed in previous articles in this series and will refine our smtp class to do just that.

If you receive my newsletter, then it's likely you have noticed that I have been sending the newsletter in HTML format this last month. Testing the HTML/SMTP code I'll present in this article. I'm going to very briefly review the base SMTP protocol, so you newbies may want to check my previous article on SMTP, http://www.kbcafe.com/articles/smtp.html.

SMTP is "the" protocol for sending email over the Internet. The protocol involves establishing a TCP/IP connection to an SMTP server and having a natural language textual conversation with the server. A textual message would contain the following body. The basic SMTP headers followed by a textual message.

```
Subject: My Subject
To: recipient@kbcafe.com
From: sender@kbcafe.com

This is the body of the message.
```

The body of the message is a set of headers with at minimum the Subject, To and From headers. Following the headers are two new lines causing one empty line. Next is the rest of the body of the message followed again by two new lines indicating the end of the message.

```
Subject: My Subject
To: recipient@kbcafe.com
From: sender@kbcafe.com
MIME-Version: 1.0
Content-Type: text/html
        charset="iso-8859-1"

<HTML>
<BODY>
<P>This is the HTML body of the message.</PL>
</BODY>
</HTML>
```

The of the HTML message has two additional headers. The MIME-Version header should be set to 1.0. The second header indicates the type of content, in this case text/html {newline}{tab}charset="iso-8859-1".

I made two modifications to the existing smtp classes that I compiled in previous articles. The first modification was to add two additional member variables to the Smtp class.

```
class Smtp : public Socket
{
public:
        Smtp(const std::string & strServer);
        virtual ~Smtp();

        void Send();
        std::string m_strContent;
        std::string m_strHtmlContent;
        std::vector<std::string> m_vstrAttachments;

        std::string m_strSender;
        std::string m_strRecipient;
        std::string m_strSubject;

};
```

The m_strHtmlContent member is the html text that is send in a message in place of the pure text. The second member, the m_vstrAttachments I added is for my next article, where I will show how to add any number and any type of attachments to outgoing messages.

```
inline void Smtp::Send()
{
        {
                std::stringstream ss;
                ss << "MAIL FROM:<" << m_strSender << ">\r\n";
                Write(ss.str());
        }

        std::string response = Response();
        if (response.find("250") == response.npos)
        {
                throw SocketException(response);
        }

        {
                std::stringstream ss;
                ss << "RCPT TO:<" << m_strRecipient << ">\r\n";
                Write(ss.str());
        }

        response = Response();
        if (response.find("250") == response.npos)
        {
                throw SocketException(response);
        }

        {
                std::stringstream ss;
                ss << "DATA\r\n";
                Write(ss.str());
        }

        response = Response();
        if (response.find("354") == response.npos)
        {
                throw SocketException(response);
        }

        if (m_strContent.empty() && !m_strHtmlContent.empty())
        {
                std::stringstream ss;
                ss << "Subject: " << m_strSubject << "\r\n"
                        << "To: " << m_strRecipient << "\r\n"
                        << "From: " << m_strSender << "\r\n"
                        << "MIME-Version: 1.0\r\nContent-Type: text/html;\r\n
        charset=\"iso-8859-1\"\r\n"
                        << "\r\n" << m_strHtmlContent << "\r\n.\r\n";
                Write(ss.str());
        }
        else
        {
                std::stringstream ss;
                ss << "Subject: " << m_strSubject << "\r\n"
                << "To: " << m_strRecipient << "\r\n"
                << "From: " << m_strSender << "\r\n"
                << "\r\n" << m_strContent << "\r\n.\r\n";
            Write(ss.str());
        }

        response = Response();
        if (response.find("250") == response.npos)
        {
            throw SocketException(response);
        }

}
```

The second set of modification to our Smtp class is in the Send method. In the send method, I check the emptiness of the m_strContent and m_strHtmlContent member variables. One or the other content member will contain the body of the message. Dependent on which, I send the appropriate headers and the text down the socket pipe.

In a future article in this series, I will show how to send attachments with pure and HTML text using the SMTP protocol.