10,957,399 members (71,180 online)

jash.liao ▾ 633 Sign out

home articles quick answers discussions features community help

Search for articles, questions, tips

Articles » General Programming » Internet / Network » Email & SMTP

SMTP Client

Jakub Piwowarczyk, 17 Jul 2010 CPOL

Rate:

★★★★★ 4.78 (47 votes)

The CSmtp class allows to send emails with attachments. It only provides the AUTH LOGIN authentication.

Download source (Windows) - 12.8 KB

Download source (Linux) - 11.6 KB

Introduction

The `CSmtp` class allows to send emails from your program. The inspiration to write the `CSmtp` class was the article: CFastSmtp - A fast and easy SMTP class... I have used the code of `CFastSmtp` and introduced the following changes:

- some bags have been removed (i.e., free memory)
- logging in with authentication has been applied (*AUTH LOGIN*)
- sending attachments has been added
- error handling has been modified
- new headlines from MIME specifications (i.e., X-Priority) have been added
- non-blocking mode has been added
- exceptions have been used
- compatibility with Linux systems has been ensured

Typical scenarios while sending emails

After successful connection to an SMTP server, our client starts the conversation with the remote SMTP server. Each line sent by the client ought to be finished by "`\r\n`". If you want to know more details, check the References: [2], [3], [4], [5], [6], [7], [8], and [9]. In [2] is described the original SMTP protocol (1982), in [4] is discussed the SMTP extensions for authentication, and the MIME specification is improved in [5]-[9]. Below there are shown typical scenarios while sending e-mails. Example 3 fails because no TLS procedures were implemented in the `CSmtp` class. If you want to add TLS, see OpenSSL. I have introduced the following notation: *S* is a remote server, *C* is our client, and *xxx* means information censured.

Example 1 - Connecting to *smtp.wp.pl* and using an incorrect login or password:

⊟ Collapse | Copy Code

```
S: 220 smtp.wp.pl ESMTP
C: EHLO: mydomain.com
S: 250-smtp.wp.pl
   250-PIPELINING
   250-AUTH=LOGIN PLAIN
   250-AUTH LOGIN PLAIN
   250-STARTTLS
   250-SIZE
   250-X-RCPTLIMIT 100
   250-8BITMIME
C: AUTH LOGIN
S: 334 VXNlcm5hbWU6
C: Kioq
S: 334 UGFzc3dvcmQ6
C: Kioq
S: 535 blad autoryzacji, niepoprawny login lub haslo / auth failure
```

Example 2 - Connecting to *smtp.wp.pl* and using the correct login and password:

⊟ Collapse | Copy Code

```
S: 220 smtp.wp.pl ESMTP
C: EHLO: mydomain.com
S: 250-smtp.wp.pl
   250-PIPELINING
   250-AUTH=LOGIN PLAIN
```

Article

Browse Code

Stats

Revisions (7)

Alternatives

Comments (142)

Add your own alternative version

Tagged as

VC6
Windows
Win32
Visual-Studio
Dev
Linux
Intermediate

Related Articles

SMTP Client with SSL/TLS

Sending complex emails in .NET 1.1

CSmtpProxyMT 1.0

Create a simple SMTP server in C#

CSPServer, State-based Protocol Server Class

Info

First Posted	**25 Aug 2008**
Views	**233,762**
Downloads	**10,975**
Bookmarked	**170 times**

Research

Four Strategies To Reduce Your Open Source Risk

```
    250-AUTH LOGIN PLAIN
    250-STARTTLS
    250-SIZE
    250-X-RCPTLIMIT 100
    250-8BITMIME
C: AUTH LOGIN
S: 334 VXNlcm5hbWU6
C: xxx
S: 334 UGFzc3dvcmQ6
C: xxx
S: 235 go ahead
C: MAIL FROM:<me@mydomain.com>
S: 250 ok
C: RCPR TO:<friend@domain.com>
S: 250 ok
C: DATA
S: 234 go ahead
C: Date: Sun, 24 Aug 2008 22:43:45
    From: JP<mail@domain.com>
    X-Mailer: The Bat! (v3.02) Professional
    Replay-to:mail@domain.com
    X-Priority: 3 (Normal)
    To:<friend@domain.com>
    Subject: The message
    MIME Version 1.0
    Content-Type: multipart/mixed; boundary="__MESSAGE__ID__54yg6f6h6y456345"


    --__MESSAGE__ID__54yg6f6h6y456345
    Content-type: text/plain; charset=US-ASCII
    Content-Transfer-Encoding: 7bit

    This is my message.

    --__MESSAGE__ID__54yg6f6h6y456345
    Content-Type: application/x-msdownload; name="test.exe"
    Content-Transfer-Encoding: base64
    Content-Disposition: attachment; filename="test.exe"

    TVqQAAMAAAAEAAAA//8AALgAAAAAAAAAAQAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
    (...)
    SU5HWFhQQURESU5HUEFERElOR1hYUEFERElOR1BBRERJTkdYWFBBRERJTkdQQURESU5HWA==
    --__MESSAGE__ID__54yg6f6h6y456345
    Content-Type: application/x-msdownload; name="test2.jpg"
    Content-Transfer-Encoding: base64
    Content-Disposition: attachment; filename="test2.jpg"

    /9j/4Sv+RXhpZgAASUkqAAgAAAAJAA8BAgAGAAAAegAAABABAgAWAAAAgAAAABIBAwABAAAA
    (...)
    A6YxR5YJJ5zUu6ZW4+NjC24E4q5Dcox5I+lRI0iWAAV9aay+lTctoYTjrml+9irRmz//2Q==

    --__MESSAGE__ID__54yg6f6h6y456345--

    .

S: 250 ok xxx qp xxx
C: QUIT
S: 221 smtp.wp.pl
```

Example 3 - Connecting to *smtp.gmail.com*:

⊟ Collapse | Copy Code

```
S: 220 mx.google.com ESMTP
    w28sm1561195uge.4
C: EHLO: mydomain.com
S: 250-mx.google.com at your service [xxx.xxx.xxx.xxx],
    250-SIZE 28311552
    250-8BITMIME
    250-STARTTLS
    250 ENHANCEDSTATUSCODES
C: AUTH LOGIN
S: 530 5.7.0 Must issue a STARTTLS command first. w28sm1561195uge.4
```

Example 4 - Connecting to *smtp.bizmail.yahoo.com* and using an incorrect login or password:

⊟ Collapse | Copy Code

```
S: 220 smtp103.biz.mail.re2.yahoo.com ESMTP
C: EHLO: mydomain.com
S: 250-smtp103.biz.mail.re2.yahoo.com
    250-AUTH LOGIN PLAIN XYMCOOKIE
    250-PIPELINING
    250-8BITMIME
C: AUTH LOGIN
S: 334 VXNlcm5hbWU6
C: dG9t
S: 334 UGFzc3dvcmQ6
C: bmVyb24xMg==
S: 535 authorization failed (#5.7.0)
```

Implementation of the CSmtp class

Implementation of the `CSmtp` class is very similar for Windows and Linux OSs. There is nothing surprising in this, because Windows uses the generally accepted Berkeley sockets application programming interface (API) [10]. The differences are shown in Table 1 (applies only to the `CSmtp` class implementation).

Table 1. Differences in implementation.

Windows	Linux
Winsock initialization needed	No Winsock initialization
Uses function `closesocket`	Uses function `close`
Uses function `ioctlsocket`	Uses function `ioctl`
Defined helpful type aliases; i.e., `SOCKET`, `SOCKADDR_IN`, `LPHOSTENT`, `LPSERVENT`, `LPIN_ADDR`, `LPSOCKADDR`	Additional types aliase should be defined

Below there are shown steps to be taken when connecting to a remote SMTP server.

1. In Windows only, initialize `Winsock2` (function: `WSAStartup`).
2. Get socket descriptor on the local machine (function: `socket`).
3. Convert port value (i.e., 25) to TCP/IP byte order (function: `htons`).
4. Obtain Internet address of the remote machine (functions: `inet_addr`, `gethostbyname`).
5. If non-blocking mode is used, set socket parameters (function: `ioctl`/`ioctlsocket`). Check necessarily what returns each function which will be called after `ioctl`/`ioctlsocket` (see next section - Using non-blocking mode).
6. Connect to the remote server (function: `connect`).
7. Introduce yourself - *EHLO <SP> <domain> <CRLF>*.
8. Send *AUTH LOGIN <CRLF>* and another command described in the section "Typical scenarios while sending the email" (functions: `send`, `recv`).
9. Finish the conversation with *QUIT <CRLF>*.
10. Close connection with remote machine (function: `close`/`closesocket`).
11. In Windows only, free `Winsock2` resources (function: `WSACleanup`).

Using non-blocking mode

In the latest version of the program, I have used a non-blocking connection. There are many strategies to implement the non-blocking mode (i.e., Select model, WSAAsyncSelect model, WSAEventSelect model, or Completion port I/O model). In my code, I have decided to use the Select model. It is not so complicated as other methods, and works efficiently with a basic connection - one client to one server. The advantages of using non-blocking mode are: the program does not suspend if the remote server stops responding, data can be sent in uneven and unequal portions. Disadvantage of this approach is its complexity. After placing the socket in non-blocking mode, the next API calls are immediately closed. Typically, these calls fail with a an error `WSAEWOULDBLOCK` (Windows) or `EINPROGRESS` (Linux), which means that the requested operation is not completed so far. Therefore, in non-blocking mode, a lot of attention should be devoted to analyze errors returned by the API functions. In Select model, we are using the `select` function [11] after calling such API functions as: `send`, `recv`, `connect`, `accept`, and others. The parameter `ndfs` in `select` is ignored in Windows, but in Linux, it is the highest-numbered file descriptor in any of the three sets (`fd_set *readfds`, `fd_set *writefds`, `fd_set *exceptfds`) plus 1. To illustrate the difference between blocking and non-blocking modes, presented here are two ways of connecting to the remote server. For greater legibility, I have only presented versions for Windows (preprocessor directives were omitted).

⊟ Collapse | Copy Code

```
/*Connecting to the remote server in blocking mode*/
SOCKET CSmtp::ConnectRemoteServer(const char *szServer,const unsigned short nPort_)
{
    unsigned short nPort = 0;
    LPSERVENT lpServEnt;
    SOCKADDR_IN sockAddr;
    unsigned long ul = 1;
    int res = 0;

    SOCKET hSocket = INVALID_SOCKET;

    if((hSocket = socket(PF_INET, SOCK_STREAM,0)) == INVALID_SOCKET)
        throw ECSmtp(ECSmtp::WSA_INVALID_SOCKET);

    if(nPort_ != 0)
        nPort = htons(nPort_);
    else
    {
        lpServEnt = getservbyname("mail", 0);
        if (lpServEnt == NULL)
            nPort = htons(25);
        else
            nPort = lpServEnt->s_port;
    }

    sockAddr.sin_family = AF_INET;
    sockAddr.sin_port = nPort;
    if((sockAddr.sin_addr.s_addr = inet_addr(szServer)) == INADDR_NONE)
    {
        LPHOSTENT host;

        host = gethostbyname(szServer);
        if (host)
            memcpy(&sockAddr.sin_addr,host->h_addr_list[0],host->h_length);
        else
        {
            closesocket(hSocket);
            throw ECSmtp(ECSmtp::WSA_GETHOSTBY_NAME_ADDR);
        }
    }

    if(connect(hSocket,(LPSOCKADDR)&sockAddr,sizeof(sockAddr)) == SOCKET_ERROR)
    {
        closesocket(hSocket);
        throw ECSmtp(ECSmtp::WSA_CONNECT);
    }

    return hSocket;
}

/*Connecting to the remote server in non-blocking mode*/
SOCKET CSmtp::ConnectRemoteServer(const char *szServer,const unsigned short nPort_)
{
    unsigned short nPort = 0;
    LPSERVENT lpServEnt;
    SOCKADDR_IN sockAddr;
    unsigned long ul = 1;
    fd_set fdwrite,fdexcept;
    timeval timeout;
    int res = 0;

    timeout.tv_sec = TIME_IN_SEC;
    timeout.tv_usec = 0;

    SOCKET hSocket = INVALID_SOCKET;

    if((hSocket = socket(PF_INET, SOCK_STREAM,0)) == INVALID_SOCKET)
        throw ECSmtp(ECSmtp::WSA_INVALID_SOCKET);

    if(nPort_ != 0)
        nPort = htons(nPort_);
    else
    {
        lpServEnt = getservbyname("mail", 0);
        if (lpServEnt == NULL)
            nPort = htons(25);
        else
            nPort = lpServEnt->s_port;
    }

    sockAddr.sin_family = AF_INET;
    sockAddr.sin_port = nPort;
    if((sockAddr.sin_addr.s_addr = inet_addr(szServer)) == INADDR_NONE)
    {
        LPHOSTENT host;

        host = gethostbyname(szServer);
        if (host)
            memcpy(&sockAddr.sin_addr,host->h_addr_list[0],host->h_length);
        else
        {
            closesocket(hSocket);
            throw ECSmtp(ECSmtp::WSA_GETHOSTBY_NAME_ADDR);
        }
    }

    // start non-blocking mode for socket:
    if(ioctlsocket(hSocket,FIONBIO, (unsigned long*)&ul) == SOCKET_ERROR)
    {
        closesocket(hSocket);
        throw ECSmtp(ECSmtp::WSA_IOCTLSOCKET);
    }

    if(connect(hSocket,(LPSOCKADDR)&sockAddr,sizeof(sockAddr)) == SOCKET_ERROR)
    {
        if(WSAGetLastError() != WSAEWOULDBLOCK)
        {
            closesocket(hSocket);
            throw ECSmtp(ECSmtp::WSA_CONNECT);
        }
    }
    else
        return hSocket;

    while(true)
    {
        FD_ZERO(&fdwrite);
        FD_ZERO(&fdexcept);

        FD_SET(hSocket,&fdwrite);
        FD_SET(hSocket,&fdexcept);

        if((res = select(hSocket+1,NULL,&fdwrite,&fdexcept,&timeout)) == SOCKET_ERROR)
        {
            closesocket(hSocket);
            throw ECSmtp(ECSmtp::WSA_SELECT);
        }

        if(!res)
        {
            closesocket(hSocket);
            throw ECSmtp(ECSmtp::SELECT_TIMEOUT);
        }

        if(res && FD_ISSET(hSocket,&fdwrite))
            break;

        if(res && FD_ISSET(hSocket,&fdexcept))
        {
            closesocket(hSocket);
            throw ECSmtp(ECSmtp::WSA_SELECT);
        }
    } // while

    FD_CLR(hSocket,&fdwrite);
    FD_CLR(hSocket,&fdexcept);

    return hSocket;
}
```

Usage

⊟ Collapse | Copy Code

```
#include "CSmtp.h"
#include <iostream>

int main()
{
    bool bError = false;

    try
    {
        CSmtp mail;

        mail.SetSMTPServer("smtp.domain.com",25);
        mail.SetLogin("***");
        mail.SetPassword("***");
        mail.SetSenderName("User");
        mail.SetSenderMail("user@domain.com");
        mail.SetReplyTo("user@domain.com");
        mail.SetSubject("The message");
        mail.AddRecipient("friend@domain2.com");
        mail.SetXPriority(XPRIORITY_NORMAL);
        mail.SetXMailer("The Bat! (v3.02) Professional");
        mail.AddMsgLine("Hello,");
        mail.AddMsgLine("");
        mail.AddMsgLine("How are you today?");
        mail.AddMsgLine("");
        mail.AddMsgLine("Regards");
        mail.AddMsgLine("--");
        mail.AddMsgLine("User");
        mail.AddAttachment("c:\\test.exe");
        mail.AddAttachment("c:\\test2.jpg");

        mail.Send();
    }
    catch(ECSmtp e)
    {
        std::cout << "Error: " << e.GetErrorText().c_str() << ".\n";
        bError = true;
    }

    if(!bError)
    {
        std::cout << "Mail was send successfully.\n";
        return 0;
    }
    else
        return 1;
}
```

Author's notes

1. If you have problems sending an email, use Visual Studio's debugger and analyze the conversation between your SMTP server and the client; perhaps, your server needs a different kind of authentication or doesn't need it at all.
2. You are not allowed to use the `CSmtp` class for spamming.

Bibliography

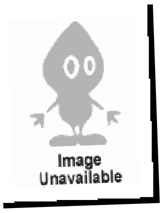
1. CFastSmtp - Fast and easy SMTP class...
2. Simple Mail Transfer Protocol RFC 821
3. Standard for the Format of ARPA Internet Text Messages RFC 822
4. SMTP Service Extension for Authentication RFC 2554
5. MIME: Format of Internet Message Bodies RFC 2045
6. MIME: Media Types RFC 2046
7. MIME: Message Header Extensions for Non-ASCII Text RFC 2047
8. MIME: Registration Procedures RFC 2048
9. MIME: Conformance Criteria and Examples RFC 2049
10. The Berkeley Sockets Application Programming Interface (API)
11. `select` function

License

Share

EMAIL

About the Author



Jakub Piwowarczyk
Engineer Technical University of Lodz
Poland

No Biography provided

Comments and Discussions

Add a Comment or Question ? **Search Comments** Go

☑ Profile popups Spacing Relaxed ▼ Noise Very High ▼ Layout Normal ▼ Per page 50 ▼ Update

First Prev Next

Using CSmtp with Managed Code new	**east75**	**7-Oct-14 21:40**
Using CSmtp without Auth [modified] new	**Jessn**	**27-Aug-14 21:59**
Fix for GetLocalHostName new	**Jessn**	**27-Aug-14 21:53**
how to send html email new	**ASERERTA@#@s**	**7-May-14 18:52**
E-mail with attachment failed. new	**Member 10655730**	**20-Mar-14 5:05**
Re: E-mail with attachment failed. new	leither908	21-Apr-14 12:01
exclamation mark (0x21) always at position 0x03b2 in received email new	**Member 10616362**	**22-Feb-14 15:07**
Re: exclamation mark (0x21) always at position 0x03b2 in received email new	Member 10616362	26-Feb-14 3:53
Attachments new	**Member 10561557**	**3-Feb-14 18:17**
Attachment error new	**Member 10561557**	**2-Feb-14 3:24**
Non-ASCII charset patch new	**S Haubenthal**	**11-Nov-13 20:02**
Error: Undefined error id. [modified] new	**Robert Hegner**	**30-Aug-13 16:38**
My vote of 5 new	**sspkmnd**	**9-Aug-13 5:35**
File is still opened after sending e-mail new	**Member 10082081**	**11-Jul-13 22:28**

My vote of 5 new	**Vivekanandm**	**8-Jul-13 15:51**
how to send an email via a proxy? new	**ASERERTA@#@s**	**9-May-13 2:00**
not able to send mail without username and password. new	**rohitnegi**	**27-Feb-13 15:47**
Got an Error new	**Sonal Dave**	**27-Feb-13 11:58**
Some Questions new	**xx0417**	**24-Jan-13 16:34**
Not working on Vista and windows 7 new	**trokfox**	**7-Jan-13 4:56**
Re: Not working on Vista and windows 7 new	beyond1977	15-Feb-14 14:01
No password? new	**Member 8688260**	**6-Dec-12 5:11**
My vote of 5 new	**gndnet**	**13-Oct-12 2:55**
portability bug w/ attachment name new	**Spike!**	**5-Sep-12 5:08**
Re: portability bug w/ attachment name new	David Johns	4-Nov-12 10:32
portability bugs w/ std::string and exceptions new	**Spike!**	**5-Sep-12 2:01**
Re: portability bugs w/ std::string and exceptions new	David Johns	4-Nov-12 10:26
My vote of 5 new	**Evren Daglioglu**	**28-Jun-12 18:45**
My vote of 4 new	**xComaWhitex**	**23-Apr-12 8:05**
Server returned error after sending MAIL FROM new	**namsaray**	**27-Jan-12 2:41**
bug fixes GetLocalHostName(), Send() new	**jerko**	**1-Dec-11 17:14**
Re: bug fixes GetLocalHostName(), Send() new	Alan P Brown	2-Dec-11 16:06
Re: bug fixes GetLocalHostName(), Send() new	Spike!	5-Sep-12 2:12
Error new	**min_2_max**	**12-May-11 22:57**
Re: Error new	min_2_max	18-May-11 12:12
Re: Error new	highersky	17-Aug-12 15:32
My vote of 3 new	**orighost**	**17-Apr-11 9:43**
Re: My vote of 3 new	Jakub Piwowarczyk	11-Dec-11 0:22
HTML body new	**nj96**	**27-Jan-11 4:53**
Re: HTML body new	Jakub Piwowarczyk	28-Jan-11 0:58
Re: HTML body new	Robert Valentino	27-Jan-12 8:12
Error:Server returned error after sending EHLO new	**caixin**	**29-Nov-10 19:33**
Re: Error:Server returned error after sending EHLO new	tmxq56	11-Jan-11 6:14
Re: Error:Server returned error after sending EHLO new	Jakub Piwowarczyk	12-Jan-11 2:25
Re: Error:Server returned error after sending EHLO new	svyeo	31-Jan-11 17:44
Re: Error:Server returned error after sending EHLO new	min_2_max	18-May-11 12:12
Re: Error:Server returned error after sending EHLO new	Ayyavu P	1-Nov-11 23:15
Re: Error:Server returned error after sending EHLO new	jerko	1-Dec-11 17:26

Re: Error:Server returned error after sending EHLO new	voidcrafter	10-Jan-13 13:56
Issue with AddMsgLine method new	**EugeneKarnygin**	**26-Nov-10 22:53**

Last Visit: 1-Jan-00 12:00 Last Update: 24-Oct-14 15:49 Refresh 1 2 3 Next »

General News Suggestion Question Bug Answer Joke Rant Admin

Use Ctrl+Left/Right to switch messages, Ctrl+Up/Down to switch threads, Ctrl+Shift+Left/Right to switch pages.

Permalink | Advertise | Privacy | Mobile
Web02 | 2.8.141022.2 | Last Updated 16 Jul 2010

Layout: fixed | fluid