TSMTP component

Properties Methods Events Tasks

Unit Smtp

declaration

```
TSMTP=class(TMailBase);
```

Description

TSMTP component processes and sends Internet mail using SMTP (Simple Mail Transfer Protocol). You can use TSMTP either to send text messages or to send text messages with multiple binary attachments.

To set your SMTP mail server. Use Server property for this purposes, to set the address of recipient, use Recipient property, the property From contains the address of sender, you can use CC property to send the same message to the multiple recipients, the Body property contains the text message itself. The Subject property holds the Subject line of your mail message.

To attach the files to your message, use Attachments property. Set Encoding property to select the method of encoding of your attachments.

Server property

Example

applies to

TMailBase class

declaration

```
property Server : string;
```

description

Address of SMTP server. Your application will make an attempt to connect to this server to send the mail. Can have format 00.00.00 or mail.somehost.com. In second case TSMTP attempts to resolve the host name using Winsock gethostbyname() function.

Recipient property

example

applies to

TSMTP component

Declaration

```
property Recipient : string;
```

description

The email address and the name of recipient.   Format is the same as for From property.

From property

example

applies to

TSMTP component

declaration

```
property From : string;
```

description

Email address and name of sender.   You can enter both email address and name of sender in this single string.   Just put | (pipe) symbol between them.   If the string consists of only one part, i.e. there is no pipe (|) symbol there, all string will be considered as Internet address.   The same rules apply to Recipient and CC properties.

CC property

Example

Applies to

TSMTP component.

Declaration

```
property CC : TStrings;
```

Carbon Copy - i.e. list of addresses and names of additional recipients.   Syntax of these lines is the same as for From property.

Body Property

example

Applies to:

TSMTP Component.

Declaration:

```
property Body : TStrings;
```

This is a body of mail message by itself.   You have to use ':=' operation to assign a value to this property.

Attachments property

Example

Applies to:

TSMTP Component.

Declaration:

```
property Attachments : TStrings;
```

Files listed in this property will be encoded and attached to the email message. If any of the files does not exist, or there is the error during encoding, exception will be raised and user will be prompted that attachment is not valid and if he wants to proceed with sending mail without attachments.

The encoding method of the attachments can be set using encoding property.

Encoding Property

applies to
TSMTP component

declaration
```
Encoding : TEncoding;
```

Encoding method of attachment.   if etUU then attachment will be UUEncoded, if etMIME then - MIME will be used.
Default is etMIME.

Subject property

Example

applies to

TSMTP component

declaration

```
property Subject : string;
```

description

A subject line of the mail message.

TMailBase class

unit
MailBase

TMailBase is the immediate ancestor of the TSMTP and TPOP3 components. All public and published properties and methods of this class are accessible from TSMTP class. Here is the declaration of TMailBase class:

declaration

```
TMailBase = class(TComponent)
private
  { Private declarations }
  FServer : string;
  FTimeOut : Integer;
  FLogFileName : string;
  FDefaultPort : word;
protected
  { Protected declarations }
  MyWSAData : WSAData;
  TheSocket : TSocket;
  ServerInAddr : PIn_Addr;
  ServerIPAddr : string;
  ThePort : word;
  Timer : TTimer;
  CurTick : Integer;
  WsInitCount : Integer;
  Log : TStrings;
  TimedOut : boolean;
  Canceled : boolean;
  ServiceName : string;
  procedure TimerOnTimer(Sender : TObject);
  procedure TimerOn;
  procedure TimerOff;
  procedure ResolveRemoteHost;
  procedure GetService;
  procedure OpenSocket;
  procedure Connect;
public
  { Public declarations }
  constructor Create(AOwner : TComponent); override;
  destructor Destroy; Override;
  procedure WriteLogFile;
  procedure Cancel; virtual;
  procedure Open; virtual;
  procedure Close; virtual;
  property LogFileName : string read FLogFileName write FLogFileName;
  property DefaultPort : word read FDefaultPort write FDefaultPort;
published
  { Published declarations }
  property Server : string read FServer write FServer;
  property TimeOut : Integer read FTimeOut write FTimeOut default 60;
end;
```

TEncoding type

Unit
SMTP

Declaration
```
TEncoding = (etUU, etMIME);
```

Description
a type of Encoding property of the TSMTP component.   Indicates the method of encoding of the binary attachments to mail messages.

properties

Attachments
Body
CC
DefaultPort
Encoding
From
LogFileName
Progress
Recipient
Server
Status
Subject
TimeOut
UseDomainAddress

TimeOut property

Example

Applies to

TMailBase class

declaration

```
property TimeOut : Integer;
```

description

Each blocking operation in TSMTP component is checked for timeout.   For example, if gethostbyname() function does not return for TimeOut seconds, exception will be raised and task will be terminated.   Default value is 60.

Example

```
SMTP1.TimeOut:=120;
```

Sets the timeout to 2 minutes.

DefaultPort property

Applies to
TMailBase class;

declaration
```
property DefaultPort : word;
```

description
Default port for smtp service.   TMailBase attempts to resolve the port number from server using getservbyname() function.   If this attempt fails, then DefaultPort property will be used to establish the connection.   Default value is 25.

example

```
OpenDialog1.Options:=[ofAllowMultiSelect];
if OpenDialog1.Execute then
  TSMTP.Attachments:=OpenDialog1.Files;
```

Selects the files and attaches them to the mail message.

example

```
SMTP1.From:='jdoe@somedomain.com|John Doe';
```

Sets email address to jdoe@somedomain.com and the name of the user to John Doe.

example

```
TSMTP1.Recipient:='jdoe@inetcom.edu|Jane Doe'
```

Sets the email address of the recipient to jdoe@inetcom.edu, and name of the recipient - Jane Doe.

example

```
with TSMTP.CC do
begin
  Clear;
  Add('jdoe@microsoft.com|John Doe');
  Add('jsmith@somenet.net|James Smith');
  Add('tadams@torfree.net');
end;
```

Sets three additional recipients of the mail message.

example

```
SMTP1.Subject:='Information you have requested';
```

example

```
SMTP1.Body:=Memo1.Lines;
```

Sets the contents of Memo1 as the body of the mail message.

example

```
SMTP1.Server:='mail.compuserve.com';
```

or

```
SMTP1.Server:='179.231.11.1';
```

In first case will be made an attempt to resolve the host name, in second case the address will be converted directly to IP.

Events

OnProgress
OnStatusChange

OnProgress event

Example

Applies to

TSMTP component

declaration

```
property OnProgress : TNotifyEvent;
```

description

Called each time the lengthy process is performed in the TSMTP methods.   Called only for processes which are controlled by TSMTP, e.g. encoding the attachments, or sending the message.   The application has to check the Progress property of TSMTP component and perform required actions.

example:

```
SMTP1.OnProgress:=SMTPProgress;
```

Sets the OnProgress event to the SMTPProgress procedure.

Here is the example of SMTPProgress procedure:

```
procedure TForm1.SMTPProgress(Sender : TObject);
begin
  ProgressBar1.Progress:=SMTP1.Progress;
end;
```

Progress property

Example

applies to

TSMTP component

declaration

```
property Progress : Integer;
```

description

Runtime and readonly.   Can be between 0..100.   If the application wants to show the progress of events occurring during sending the mail, it has to set the OnProgress event of the TSMTP instance and use Progress property.

OnStatusChange event

example

Applies to

TSMTP component

declaration

```
property OnStatusChange : TNotifyEvent;
```

description

This event allows you to check the status of your application. You can assign this event to your SMTP component instance and trigger the current status using Status property of TSMTP component. This property does not appear in object inspector, because it is not declared as published, but it is accessible because it is declared as public. Type of Status property is TStatus.

TStatus type

Unit

SMTP

declaration

```
  TStatus = (msIdle,msConnecting,msLogIn,msLogOut,
             msHeaders,msEnvelope,msBody,msAttachment,
             msCancel,msEnCode);
```

All values are self explanatory.

example

```
procedure TForm1.SMTPStatusChange(Sender : TObject);
var
  s : string;
begin
  case Mailer.Status of
   msIdle : s:='';
   msLogIn : s:='Logging In';
   msLogOut : s:='Logging Out';
   msHeaders : s:='Sending headers';
   msEnvelope : s:='Sending commands';
   msBody : s:='Sending message body';
   msAttachment : s:='Sending attachment(s)';
   msCancel : s:='Canceled';
   msEnCode : s:='Encoding the attachment(s)';
  end;
  StatusBar.Caption:=s;
end;
```

Status property

example

applies to

TSMTP component

declaration

```
property Status : TStatus;
```

description

Run-time and Read-only.   The application should use this property to detect the current status of the process, e.g. in your OnStatusChage event handler.

LogFileName property

Applies to
TMailBase class

declaration
```
property LogFileName : string;
```

description
Application should use the LogFileName property for debugging purposes. It can call WriteLogFile public event to write the contents of the Log : TStringList object. WriteLogFile method called automatically in the destructor of TMailBase class. If LogFileName is blank, WriteLogFile does nothing.

WriteLogFile method

Applies to
TMailBase class

declaration
```
procedure WriteLogFile;
```

description
This method is provided for debug purposes.   It will work only if the LogFileName property   is not blank.   You can call this method after completion of your task to write log file and find out more about the conversation between your application and SMTP server.

Methods

Cancel
Close
LogIn
LogOut
Open
Send
WriteLogFile

Open method

applies to
TMailBase class

declaration
```
procedure Open; virtual;
```

description
performs the following tasks:

1. Runs calls Winsock WSAStartup procedure which initializes Winsock, then increments an internal counter of WSAStartup calls.   This counter is used when calling WSACleanUp procedure in the destructor of TMailBase component.

2. Tries to resolve remote host Internet address based on Server property value;

3. Tries to find the port number for SMTP service.   If fails, uses the default port set in DefaultPort property, (normally 25) if DefaultPort=0, then raises exception;

3. Creates socket

TSMTP overrides this method just to call the OnStatusChange event.

Close method

applies to
TMailBase class

declaration
```
procedure Close;
```

description
Closes the connection by closing the socket.   In the normal circumstances must be used after LogOut method or if by some reason operation has been canceled or timed out.

LogOut method

applies to
TSMTP component

declaration
```
procedure LogOut;
```

description
Sends to the Server 'QUIT' command, after this server must close the connection.   If there are no error messages, displays notification that mail has been sent.

LogIn method

applies to

TSMTP component

declaration

```
procedure LogIn;
```

description

Connects the socket to the Server using Winsock Connect function, and sends HELO message to the server.

Cancel method

applies to
TMailBase class;

declaration
```
procedure Cancel; virtual;
```

description
Cancels current operation.   TSMTP overrides this method to perform some specific tasks, like canceling encoding and updating the status, by calling OnStatusChange method.

Send method

Applies to
TSMTP component

declaration
```
procedure Send;
```

description
Performs the following tasks:
1. Encodes attachments, if there are any;
2. Sends the envelope information, i.e. information about recipient and sender,
3. Sends headers.   Headers are the part of mail message, but they are subject of some agreements. See rfc822 for more info,
4. Sends message body.

Using TSMTP component

TSMTP reference

Use TSMTP component to send internet mail.   Here is the example of sending single text message.   In this example it is presumed that the published properties have been already set.

```
with TSMTP1 do
try
  Open;
  LogIn;
  Send;
  LogOut;
finally
  Close;
end;
```

Here is the example of sending multiple messages:

```
with TSMTP1 do
try
  Open;
  LogIn;
  for i:=1 to NumOfMailMessages do
  begin
    From:=SenderAddresses[i];
    Recipient:=RecipientAddresses[i];
    .
    .
    Send;
  end;
finally
  close;
end;
```

UseDomainAddress property

applies to
TSMTP component

declaration
```
property UseDomainName : boolean;
```

description
In the LogIn method TSMTP sends *HELO* command to the SMTP server.   Some servers require that this command is followed by the senders domain name, e.g. if sender's email address is jdoe@somedomain.com, then some servers require to receive the following greeting:

```
HELO somedomain.com
```

Some servers just don't care about it and they accept both variants, but there is a category of servers do not like when HELO is followed by its own domain name.   This happens not very frequently, so probably you even never need to modify this property, which is set by default to true.