



TPOP3 Component

[Properties](#)

[Methods](#)

[Events](#)

[Tasks](#)

unit

pop3

declaration

```
TPOP3=class(TMailBase)
```

description

TPOP3 is component receives the internet mail using Post Office Protocol version 3.

To specify your POP3 server IP or DN, set Server property. To access your account, you also will need to set UserName and Password properties.

To connect to POP3 server, call open method. To log in to the server and get information about received mail, call LogIn and GetStatistics methods. To retrieve received messages, use GetMessages method. To close connection with server, use LogOut and to close the socket, which is connected to mail server, call close.

Methods

[Cancel](#)

[Close](#)

[DeleteMessages](#)

[GetMessages](#)

[GetStatistics](#)

[LogIn](#)

[LogOut](#)

[Open](#)

[WriteLogFile](#)

Events

OnStatusChange

OnProgress



Using the TPOP3 component

[TPOP3 Reference](#)

Here is a sample session how to receive the mail using TPOP3 component:

To check your mailbox, and retrieve the messages, follow the following steps:

```
with POP3 do
try
  Open;
  LogIn;
  GetStatistics;
  if TotalMessages>0 then GetMessages;
  LogOut;
finally
  Close;
end;
```

The bodies of your messages will be stored into the [MailMessages](#) field of your POP3 object.

for more information have a look at pop3mail.dpr project which is included with this package.

OnProgress event

example

Applies to

TPOP3 component

declaration

```
property OnProgress : TNotifyEvent;
```

description

Called when retrieving large messages. TPOP3 first sends the LIST command to the server, then parses the received string, then starts message retrieving. During the retrieving process it calculates the percentage of the retrieved bytes, sets this percentage to the Progress property and calls OnProgress method. It means the application can display the progress of the lengthy process, as retrieving long message.

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 for POP3 is 110.

WriteLogFile method

Applies to

IMailBase 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 POP3 server.

TotalMessages Property

[Example](#)

applies to

TPOP3 component

declaration

```
TotalMessages : Integer;
```

description

Runtime and readonly. Contains information about the number of messages received.

GetStatistics Method

applies to

TPOP3 component

declaration

```
procedure GetStatistics;
```

description

Gets information about the messages in the maildrop. Fills the values of TotalMessages and TotalOctets. Called each time the application calls GetMessages or DeleteMessages methods but it is located in the public part of the TPOP3 class to allow users to retrieve information before retrieving messages.

UserName Property

Example

applies to

TPOP3 component

declaration

```
UserName : string;
```

description

Name of the user in the POP3 server account. With Password gives the user the access to POP3 server.

Example

```
POP31.UserName:='jdoe';  
POP31.Password:='blahblah';  
POP31.Login;
```

Status Property

example

applies to

TPOP3 component

declaration

Status : TPOP3Status;

description

Runtime and readonly. Indicates the status of the current operation.

TotalOctets Property

applies to

IPQP3 component

declaration

```
TotalOctets : LongInt;
```

description

Total number of bytes in all received messages

Properties

DefaultPort

Password

Progress

Server

Status

TimeOut

TotalMessages

TotalOctets

UserName

Example

```
if POP31.TotalMessages=0 then
  StatusBar.Caption:='You have no new messages'
else
  StatusBar.Caption:=Concat('You have ',IntToStr(POP31.TotalMessages),'
messages.');
```

Password Property

Example

Applies to

TPOP3 component

declaration

```
Password : string;
```

description

This is a password which is required to login into the POP3 server.

Example

TPOP3.Close

DeleteMessages Method

Example

applies to

TPOP3 component

declaration

```
procedure DeleteMessages;
```

description

Marks messages on the Server for deletion. Actual deletion occurs when you run LogOut method.

Server Property

Example

applies to

TPOP3 component

declaration

```
property Server : string;
```

description

Address of POP3 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 TPOP3 attempts to resolve the host name using Winsock *gethostbyname()* function.

Example

```
POP31.Server:='mail.compuserve.com';
```

or

```
POP31.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.

TimeOut Property

[Example](#)

applies to

TMailBase class

declaration

```
property TimeOut : Integer;
```

description

Each blocking operation in TPOP3 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

```
TPOP31.TimeOut:=30;
```

sets timeout to 30 seconds.

Cancel Method

[Example](#)

applies to

TPOP3 component

declaration

```
procedure Cancel;
```

description

Cancels current operation

Example

```
function TForm1.CancelButtonClick(Sender : TObject);  
begin  
    POP31.Cancel;  
end;
```


Open Method

applies to

TMailBase class

declaration

```
procedure Open; virtual;
```

description

performs the following tasks:

1. Runs calls Winsock WSASStartup procedure which initializes Winsock, then increments an internal counter of WSASStartup 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 POP3 service. If fails, uses the default port set in DefaultPort property, (normally 110) if DefaultPort=0, then raises exception;
3. Creates socket

TPOP3 overrides this method just to call the OnStatusChange event.

LogIn Method

applies to

TPOP3 component

declaration

```
procedure LogIn;
```

description

Connects to Server using Winsock Connect function, and sends UserName and Password to the server.

example

```
POP3.OnProgress:=POP3Progress;
```

Sets the OnProgress event to the POP3Progress procedure.

Here is the example of POP3Progress procedure:

```
procedure TForm1.POP3Progress(Sender : TObject);  
begin  
    ProgressBar1.Progress:=POP3.Progress;  
end;
```

Close Method

Example

applies to

TPOP3 component

declaration

```
procedure Close;
```

description

Closes the connection by closing the socket. Must be used after LogOut method.

Example

```
if not LeaveOnServer then  
  POP31.DeleteMessages;
```

Progress property

example

applies to

TPOP3 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 when receiving the mail, it has to set the OnProgress event of the TPOP3 instance and use Progress property.

OnStatusChange Event

[Example](#)

Applies to

TPOP3 component

declaration

```
property OnStatusChange : TNotifyEvent;
```

description

This event allows you to check the status of your application. You can assign this event to your POP3 component instance and trigger the current status using Status property of TPOP3 component. Status property does not appear in the object inspector, because it is not declared as published, but it is accessible because it is declared as public. Type of Status property is TPOP3Status.

TPOP3Status Type

Unit

POP3

declaration

```
TPOP3Status = (psIdle,psLogIn,psResolving,psConnecting,  
              psRetrieving,psDeleting,psLogOut,psCancel,psTimeOut);
```

All values are self explanatory.

Example

```
procedure TForm1.SMTP1StatusChange(Sender : TObject);
var
  s : string;
begin
  case POP3.Status of
    psIdle : s:='';
    psResolving : s:='Resolving remote host';
    psConnecting : s:='Connecting to server';
    psLogIn : s:='Logging In';
    psLogOut : s:='Logging Out';
    psRetrieving : s:='Retrieving Message(s)';
    psDeleting : s:='Deleting Message(s)';
    psCancel : s:='Canceled';
    psTimeOut : s:='Timed Out';
  end;
  StatusBar.Caption:=s;
end;
```

LogOut Method

applies to

TPQP3 component

declaration

```
procedure LogOut;
```

description

Sends to the server 'QUIT' command, after this server must close the connection.

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 TPOP3 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;
```


GetMessages Method

applies to

TPOP3 component;

declaration

```
procedure GetMessages;
```

description

Retrieves all messages from maildrop and fills MailMessages field of TPOP3 component. First calls GetStatistics method and then retrieves message bodies.

MailMessages

applies to

TPOP3 component

declaration

```
MailMessages : TStringList;
```

This is very important part of TPOP3 component. This TStringList object contains all messages. Messages are of type TMailMessage and they are attached to the strings as objects.

TMailMessage class

unit TPOP3

declaration

```
TMailMessage = class
protected
  Headers : TStrings;
  TempLines : TStrings;
public
  From : string;
  Subject : string;
  Size : LongInt;
  Body : TMemoryStream;
  constructor Create;
  destructor Destroy; override;
  procedure FillBody(AStream : TStream);
  procedure FillFields;
end;
```

TMailMessage class objects are located in the MailMessages lines, as objects.

When TPOP3 receives the message, it creates instance of TMailMessage object, assigns to its body field the received MemoryStream, calling FillBody, then calls FillFields, which retrieves the header information from the Body and fills the fields From, Subject and Size.

Then the instance of TMailMessage class is assigned to the TPOP3.MailMessages field. The following example illustrates how to save the bodies of all received messages into the separate files:

```
Procedure SaveAllMessages;
var
  i : Integer;
begin
  for i:=0 to MailMessages.Count-1 do
    TMailMessage(TPOP3.Messages[i]).SaveToFile(MakeFileName(i));
end;
```

Note that count of messages is zero-based.

