NMNNTP unit

The NMNNTP unit contains the TNMNNTP Component and it's related types and objects.

Components
TNMNNTP

Objects
TPostRecordType

Types
TCacheMode
TGroupRetrievedCacheEvent
TGroupRetrievedEvent
THeaderCacheEvent

TPostRecordType object

See also Properties

Unit

NMNNTP

Description

The TPostRecord type is used for storing NNTP header information. It need not necessarily be for posting a message, but also for a retrieved message.

See also

TNMNNTP.HeaderRecord property

Properties

Run-time only
Key properties

- PrAppName
- PrArticleId
- PrDistribution
- PrFromAddress
- PrNewsGroups
- PrReplyTo
- PrSubject
- PrTimeDate

PrAppName property

Applies to
TPostRecordType object

Declaration
```
property PrAppName: string;
```

Description
The **PrAppName** property specifies the name of the application that posted the news message.

Scope: Published
Accessability: Runtime, designtime

PrArticleId property

Applies to
TPostRecordType object

Declaration
```
property PrArticleId: integer;
```

Description
The **PrArticleId** property contains the message ID number for the current message.

Scope: Published
Accessability: Runtime, read-only

PrDistribution property

Applies to
TPostRecordType object

Declaration
```
property PrDistribution: string;
```

Description
The **PrDistribuition** property is used to alter the distribution scope of the message. It is a comma separated list similar to the "Newsgroups" line. User subscriptions are still controlled by "Newsgroups", but the message is sent to all systems subscribing to the newsgroups on the "Distribution" line in addition to the "Newsgroups" line. For the message to be transmitted, the receiving site must normally receive one of the specified newsgroups AND must receive one of the specified distributions.

Scope: Published
Accessability: Runtime, designtime

PrFromAddress property

Applies to
TPostRecordType object

Declaration
```
property PrFromAddress: string;
```

Description
The **PrFromAddress** property specifies the E-Mail addressof the person that posted the current news article.

Scope: Published
Accessability: Runtime, designtime

PrNewsGroups property

Applies to
TPostRecordType object

Declaration
```
property PrNewsGroups: string;
```

Description
The **PrNewsGroups** property specifies the news groups(s) that the current message is posted in.

Scope: Published
Accessability: Runtime, designtime

PrReplyTo property

Applies to
TPostRecordType object

Declaration
```
property PrReplyTo: string;
```

Description
The **PrReplyTo** property contains the E-Mail address that can be used to reply to the person who posted the current message.

Scope: Published
Accessability: Runtime, designtime

PrSubject property

Applies to
TPostRecordType object

Declaration
```
property PrSubject: string;
```

Description
The **PrSubject** property contains the subject line from the news article

Scope: Published
Accessability: Runtime, designtime

PrTimeDate property

Applies to
TPostRecordType object

Declaration
```
property PrTimeDate: string;
```

Description
The **PrTimeDate** property contains the date and time that the message was posted.

Scope: Published
Accessability: Runtime, designtime

TNMNNTP component

Properties Methods Events Tasks

Unit

NMNNTP

Description

The TNMNNTP component is used for the reading and posting of internet news articles on internet and intranet news servers.

TNMNNTP Properties
TNMNNTP
Legend

In TNMNNTP

AttachFilePath
Attachments
Body
CacheMode
CurrentArticle
GroupList
Header
HeaderRecord
HiMessage
LoMessage
NewsDir
ParseAttachments
Password
PostAttachments
PostBody
PostHeader
Posting
PostRecord
SelectedGroup
UserId

Derived from TPowersock

About
BeenCanceled
BeenTimedOut
BytesRecvd
BytesSent
BytesTotal
Connected
Handle
Host
LastErrorNo
LocalIP
Port
Proxy
ProxyPort

RemoteIP

ReplyNumber

ReportLevel

Status

TimeOut

TransactionReply

WSAInfo

Derived from TComponent

ComObject

ComponentCount

ComponentIndex

Components

ComponentState

ComponentStyle

DesignInfo

Owner

Tag

VCLComObject

TNMNNTP Methods
TNMNNTP
Legend

In TNMNNTP
- GetArticle
  - GetArticleBody
  - GetArticleHeader
  - GetArticleList
- GetGroupList
- PostArticle
- SetGroup

Derived from TPowersock
- Abort
- Accept
  - Cancel
- CaptureFile
- CaptureStream
- CaptureString
  - CertifyConnect
- Connect
  - Create
  - Destroy
- Disconnect
  - FilterHeader
  - GetLocalAddress
  - GetPortstring
- Listen
- read
- ReadLn
  - RequestCloseSocket
  - SendBuffer
- SendFile
- SendStream
- Transaction
- write
- writeln

Derived from TComponent
  - DestroyComponents
  - Destroying
  - FindComponent
  - FreeNotification
- FreeOnRelease
- GetParentComponent
- HasParent
- InsertComponent
- RemoveComponent
- SafeCallException

Derived from TPersistent
- Assign
- GetNamePath

Derived from TObject

ClassInfo
ClassName
ClassNameIs
ClassParent
ClassType
CleanupInstance
DefaultHandler
Dispatch
FieldAddress
Free
FreeInstance
GetInterface
GetInterfaceEntry
GetInterfaceTable
InheritsFrom
InitInstance
InstanceSize
MethodAddress
MethodName
NewInstance

TNMNNTP Events
TNMNNTP
Legend

In TNMNNTP

- OnAbort
- OnArticle
  - OnArticleCacheUpdate
  - OnAuthenticationFailed
  - OnAuthenticationNeeded
  - OnBody
  - OnBodyCacheUpdate
  - OnGroupListCacheUpdate
  - OnGroupListUpdate
  - OnGroupSelect
  - OnGroupSelectRequired
  - OnHeader
  - OnHeaderCacheUpdate
  - OnHeaderList
  - OnHeaderListCacheUpdate
  - OnInvalidArticle
  - OnPosted
- OnPostFailed

Derived from TPowersock

- OnAccept
- OnConnect
  - OnConnectionFailed
- OnConnectionRequired
- OnDisconnect
- OnError
- OnHostResolved
- OnInvalidHost
- OnPacketRecvd
- OnPacketSent
- OnRead
- OnStatus

About the TNMNNTP component

TNMNNTP reference

Purpose

The Purpose of the TNMNNTP component is to read and post internet news to Internet news servers using the Network News Transfer Protocol. Use of this component requires a 32 bit Winsock stack, WSOCK32.DLL, which is available from various vendors, and is included with Windows 95.

RFC: RFC 977

Tasks

Before you can perform any of the key functions of the TNMNNTP component, you must first connect to the news host. This is done by first setting the **Host** property to a valid news server. Then, call the **Connect** method to connect to the server.

Getting a list of News Groups:
A list of the news groups on a given news server can be obtained by calling the **GetGroupList** method.

Selecting a newsgroup to read articles from:
To select a newsgroup to read articles from, call the **SetGroup** method, passing the name of the newsgroup you wish to read news from as the parameter.

Reading Internet News Articles:
To read internet news articles once you have selected a group, just call the **GetArticle** method to retrieve an article. Then, by viewing the value of the **Body** property and the **Header** property, you can read the selected article.

Posting internet news articles:
Posting internet news articles is done by calling filling the PostBody property with the body of the message, and the PostHeader property with the header of the message, and then calling the **PostArticle** method.

AttachFilePath property

See also Example

Applies to

TNMNNTP component

Declaration

```
property AttachFilePath: string;
```

Description

The **AttachFilePath** property specifies the location to save files attached to news articles that are listed in the **Attachments** property.

Default:
The default value for this property is blank, which will save file attachments into the applications directory.

Scope: Published
Accessability: Runtime, designtime

Notes:
If an invalid path is specified for AttachFilePath, the directory in which the application resides will be used as the AttachFilePath.
Files are only stored in the **AttachFilePath** directory if the **ParseAttachments** property is TRUE.

See also

Attachments property
ParseAttachments property

Example

To recreate this example, you will need to create a new Delphi application.

Place 4 TEdits, 7 TButtons, 3 TMemos, 2 TListBoxes, 2 TLabels, a TCheckBox, and a TNMNNTP on the form.

Component Descriptions:
Edit1: news host or IP address
Edit2: Poster of the article
Edit3: Subject of the article
Edit4: Time/Date article was posted
Button1: Connect/Disconnect
Button2: Get Article
Button3: Get Article Body
Button4: Get Article Header
Button5: Get Article (Header) List
Button6: Abort current operation
Button7: Change attach file path
Memo1: list of files attached to article (if any)
Memo2: Article header
Memo3: Article Body
ListBox1: Newsgroup list
ListBox2: Article list
CheckBox1: ParseAttachments On/Off
Label1: Display attach file path
Label2: Current Article Number

Be sure to add **filectrl** to the **uses** clause of unit1. The top of your **interface** should look something like this:

```
implementation

{$R *.DFM}

uses
  filectrl;
```

Insert the following code into Button1's OnClick event:

```
procedure TForm1.Button1Click(Sender: TObject);
begin
  if NMNNTP1.Connected then
    NMNNTP1.Disconnect
  else
    begin
      NMNNTP1.Host := Edit1.Text;
      NMNNTP1.Connect;
    end;
end;
```

When Button1 is clicked, if the **Connected** property is true, a connection is present, and the **Disconnect** method closes the connection. If there is no connection present, the **Host** property is set to the value of the text in Edit1, and the **Connect** method attempts to establish a connection with the remote news host.

Insert the following code into NMNNTP1's OnConnect event:

```
procedure TForm1.NMNNTP1Connect(Sender: TObject);
var
  I: Integer;
begin
  StatusBar1.SimpleText := 'Connected';
  NMNNTP1.GetGroupList;
  for I := 0 to (NMNNTP1.GroupList.Count - 1) do
    ListBox1.Items.Add(NthWord(NMNNTP1.GroupList[I], ' ', 1));
end;
```

When a connection is established with the remote news host, the **OnConnect** event notifies the user of the connection by updating the status bar. The **GetGroupList** method is called to retrieve a list of the available newsgroups on the host. The **GroupList** property is then iterated through, and the **NthWord** function is used to parse out the first word in each string, which happens to be the name of the newsgroup. This string is then added to ListBox1.

Insert the following code into ListBox1's OnClick event:

```
procedure TForm1.ListBox1Click(Sender: TObject);
begin
  if ListBox1.ItemIndex <> -1 then
    begin
      ListBox2.Clear;
      NMNNTP1.SetGroup(ListBox1.Items[ListBox1.ItemIndex]);
      Caption := 'Reading - '+NMNNTP1.SelectedGroup+' Lo: '+IntToStr(NMNNTP1.LoMessage)+' Hi:
'+IntToStr(NMNNTP1.HiMessage);
    end;
end;
```

When ListBox1 is clicked, ListBox2 is cleared, and the **SetGroup** method is used to select the group that was selected from ListBox1. The Caption of the form is then modified to display the name of the newsgroup in the **SelectedGroup** property, and also the first and last message numbers in the **LoMessage** and **HiMessage** properties, respectively.

Insert the following code into NMNNTP1's OnDisconnect event:

```
procedure TForm1.NMNNTP1Disconnect(Sender: TObject);
begin
  StatusBar1.SimpleText := 'Disconnected';
end;
```

When the client disconnects from the remote news host, the **OnDisconnect** event is called. In this instance, StatusBar1 is updated to display that the client has disconnected.

Insert the following code into Button5's OnClick event:

```
procedure TForm1.Button5Click(Sender: TObject);
begin
  NMNNTP1.GetArticleList(TRUE, 0);
```

```
end;
```

When Button5 is clicked, the **GetArticleList** method is called to get a listing of article in the current newsgroup. Since the All parameter was passed TRUE, all of the messages available in the selected group are listed.

Insert the following code into NMNNTP1's OnHeaderList event:

```
procedure TForm1.NMNNTP1HeaderList(Sender: TObject);
begin
  ListBox2.Items.Add('['+IntToStr(NMNNTP1.HeaderRecord.PrArticleId)+']Subject:
'+NMNNTP1.HeaderRecord.PrSubject);
end;
```

When the **GetArticleList** method is called, for each article listed, the **OnHeaderList** event is called. Here, the **ArticleID** property and the **PrSubject** property of the **HeaderRecord** property are added to ListBox2 for display.

Insert the following code into Button6's OnClick event:

```
procedure TForm1.Button6Click(Sender: TObject);
begin
  NMNNTP1.Abort;
end;
```

When Button6 is clicked, the **Abort** method is called, cancelling the current operation.

Insert the following code into NMNNTP1's OnAbort event:

```
procedure TForm1.NMNNTP1Abort(Sender: TObject);
begin
  ShowMessage('Aborted');
end;
```

When the **Abort** method is called, the **OnAbort** event is called. In this case, the ShowMessage procedure is used to display a dialog box stating that the current operation was aborted.

Insert the following code into Button7's OnClick event:

```
procedure TForm1.Button7Click(Sender: TObject);
var
  S: String;
begin
  if SelectDirectory(S, [sdAllowCreate, sdPerformCreate, sdPrompt], 0) then
    begin
      Label1.Caption := S;
      NMNNTP1.AttachFilePath := S;
    end;
end;
```

When Button7 is clicked, the SelectDirectory function (located in the **filectrl** unit) is used to allow the user to select a directory. If a directory is selected and the Ok button is clicked, the caption of Label1 is

changed to display the directory selected, and the **AttachFilePath** property is set to the directory chosen.

Insert the following code into CheckBox1's OnClick event:

```
procedure TForm1.CheckBox1Click(Sender: TObject);
begin
  if CheckBox1.Checked then
    NMNNTP1.ParseAttachments := TRUE
  else
    NMNNTP1.ParseAttachments := FALSE;
end;
```

When CheckBox1 is clicked, if the checkbox is checked, then the **ParseAttachments** property is set to TRUE, so attachments will be parsed out of the article body and saved on disk. If the checkbox is not checked, the value is instead false, and file attachments are left in the article body.

Insert the following code into Button4's OnClick event:

```
procedure TForm1.Button4Click(Sender: TObject);
var
  S: String;
  I: Integer;
begin
  if InputQuery('Retrieve article header', 'Which article header to retrieve', S) then
    begin
      I := StrToIntDef(S, -1);
      if I <> -1 then
        NMNNTP1.GetArticleHeader(I);
    end;
end;
```

When Button4 is clicked, the InputQuery function is used to retrieve the article id of the article which to retrieve the header of. If the Ok button is clicked, and the text entered is indeed a number, the **GetArticleHeader** method is called to retrieve the header for the selected article.

Insert the following code into NMNNTP1's OnHeader event:

```
procedure TForm1.NMNNTP1Header(Sender: TObject);
begin
  Memo2.Text := NMNNTP1.Header.Text;
  Edit2.Text := NMNNTP1.HeaderRecord.PrFromAddress;
  Edit3.Text := NMNNTP1.HeaderRecord.PrSubject;
  Edit4.Text := NMNNTP1.HeaderRecord.PrTimeDate;
end;
```

When a header is retrieved from the remote news host, the **OnHeader** event is called. In this instance, the Text in Memo2 is set to the value of the text in the **Header** property. Edit2 is set to the value of the **HeaderRecord.PrFromAddress** property, which contains the poster of the news article. Edit3 is set to the value of the **HeaderRecord.PrSubject** property, which contains the subject line of the posted article. Edit4 is set to the value of the **HeaderRecord.PrTimeDate** property, which contains the time and date that the article was posted.

Insert the following code into Button2's OnClick event:

```
procedure TForm1.Button2Click(Sender: TObject);
var
  S: String;
  I: Integer;
begin
  if InputQuery('Retrieve article', 'Which article to retrieve', S) then
    begin
      I := StrToIntDef(S, -1);
      if I <> -1 then
        NMNNTP1.GetArticle(I);
    end;
end;
```

When Button2 is clicked, the InputQuery function is used to retrieve the article id of the article which to retrieve. If the Ok button is clicked, and the text entered is a valid article id number, the **GetArticle** method retrieves the article selected.

Insert the following code into NMNNTP1's OnInvalidArticle event:

```
procedure TForm1.NMNNTP1InvalidArticle(Sender: TObject);
begin
  ShowMessage('Invalid article number');
end;
```

When a specified article ID is invalid, the **OnInvalidArticle** event is called. Here, the ShowMessage procedure is used to inform the user that the specified article ID is invalid.

Insert the following code into Button3's OnClick event:

```
procedure TForm1.Button3Click(Sender: TObject);
var
  S: String;
  I: Integer;
begin
  if InputQuery('Retrieve article body', 'Which article body to retrieve', S) then
    begin
      I := StrToIntDef(S, -1);
      if I <> -1 then
        NMNNTP1.GetArticleBody(I);
    end;
end;
```

When Button3 is clicked, the InputQuery function is used to retrieve the article id of the article which to retrieve the body for. If the Ok button is clicked, and the text entered is a valid article ID, the **GetArticleBody** method is called to retrieve the selected article body.

Insert the following code into NMNNTP1's OnArticle event:

```
procedure TForm1.NMNNTP1Article(Sender: TObject);
begin
  Memo3.Text := NMNNTP1.Body.Text;
```

```
  Memo2.Text := NMNNTP1.Header.Text;
  Memo1.Text := NMNNTP1.Attachments.Text;
  Edit2.Text := NMNNTP1.HeaderRecord.PrFromAddress;
  Edit3.Text := NMNNTP1.HeaderRecord.PrSubject;
  Edit4.Text := NMNNTP1.HeaderRecord.PrTimeDate;
  Label2.Caption := IntToStr(NMNNTP1.CurrentArticle);
end;
```

When an article is retrieved from the remote news host, the **OnArticle** event is called. Here, Memo3.Text is set to the value of the **Body** property, to display the body of the retrieved article. Memo2 is set to the value of the **Header** property to display the header of the retrieved article. Memo1 is set to the value of the **Attachments** property, which contains a list of the files attached to the article (if any). Edit2 is set to the value of the **HeaderRecord.PrFromAddress** property, which contains the poster of the news article. Edit3 is set to the value of the **HeaderRecord.PrSubject** property, which contains the subject line of the posted article. Edit4 is set to the value of the **HeaderRecord.PrTimeDate** property, which contains the time and date that the article was posted. The caption of Label2 is changed to display the id of the last retrieved article, contained in the **CurrentArticle** property.

Insert the following code into NMNNTP1's OnBody event:

```
procedure TForm1.NMNNTP1Body(Sender: TObject);
begin
  Memo2.Text := NMNNTP1.Body.Text;
  Memo1.Text := NMNNTP1.Attachments.Text;
end;
```

When the **GetArticleBody** method is called, the **OnBody** event is called when the body has been retrieved from the remote news host. Here, the text of the **Body** property is displayed in Memo2, and the list of files attached to the body (if any) are displayed in Memo1.

Insert the following code into NMNNTP1's OnGroupSelectRequired event:

```
procedure TForm1.NMNNTP1GroupSelectRequired(var Handled: Boolean);
var
  S: String;
begin
  if InputQuery('Group selection required', 'Newsgroup: ', S) then
    begin
      NMNNTP1.SetGroup(S);
      Handled := TRUE;
    end;
end;
```

When a method is called that requires a newsgroup to be selected (such as the **GetArticleList** method), and there is no group currently selected, the **OnGroupSelectRequired** event is called. In this case, the InputQuery function is used to obtain the name of a newsgroup. If the Ok button is clicked, the **SetGroup** method is called to set the newsgroup to the group name entered. The Handled parameter is also set to TRUE. If the Cancel button were clicked, an exception would be raised due to the Handled parameter remaining the default of FALSE.

Attachments property

See also Example

Applies to
TNMNNTP component

Declaration
```
property Attachments: TStringList;
```

Description
The **Attachments** property is a list of files attached to the current news article that has been received from the remote host. These files are only decoded and stored on disk in the directory specified by the **AttachFIlePath** property if the **ParseAttachments** property is TRUE.

Scope: Public
Accessability: Runtime, Read-Only

Note:
The Attachments property is set after a call to the GetArticle method.

See also

AttachFilePath property
ParseAttachments property

Body property

See also Example

Applies to

TNMNNTP component

Declaration

```
property Body: TExStringList;
```

Description

The **Body** property contains the body of the current news article retrieved from the remote host.

Scope: Public
Accessability: Runtime, Read-only

Note:
The Body property is set after a call to the **GetArticle** or **GetArticleBody** method.
If the **GetArticleBody** method is called, even if the **ParseAttachments** property is set to TRUE, any files attached to the news article are not parsed out, and are returned in the article's body.

See also

GetArticle method
GetArticleBody method
Header property
ParseAttachments property

CacheMode property

See also

Applies to

TNMNNTP component

Declaration

```
property CacheMode: TCacheMode;
```

Description

The **CacheMode** property specifies the type of news caching to use.

Default: cmMixed

Range: The following values are defined by the TCacheMode type:
cmMixed: When articles, group lists, headers, or header lists are requested, they are downloaded from the remote host. They are also stored on disk in this mode.

cmLocal: When articles, group lists, headers, or header lists are requested, they are loaded from the local disk. If they do not exist on disk, the **OnInvalidArticle** event will be called, just as if the article did not exist on a remote host.

cmRemote: When articles, group lists, headers, or header lists are requested, they are downloaded from the remote host and are not saved to disk.

Notes:
When CacheMode is cmMixed, when articles are retrieved from the remote host, they are stored on the local disk. The default caching routines may be overriden in this case by writing handlers for the following events:

OnArticleCacheUpdate
OnBodyCacheUpdate
OnGroupListCacheUpdate
OnHeaderCacheUpdate
OnHeaderListCacheUpdate

NOTE
At the present, the caching functions of TNMNNTP are disabled. All messages are downloaded from the remote host regardless of the CacheMode that is selected.

See also

NewsDir property
OnArticleCacheUpdate event
OnBodyCacheUpdate event
OnGroupListCacheUpdate event
OnHeaderCacheUpdate event
OnHeaderListCacheUpdate event

CurrentArticle property

See also Example

Applies to
TNMNNTP component

Declaration
```
property CurrentArticle: integer;
```

Description
The **CurrentArticle** property specifies the last article that was downloaded from the news host. Typically, this is the article number of the article contained in the **Body** and **Header** properties.

Scope: Public
Accessability: Runtime, Read-Only

Notes:
This property is changed by a successful call to the **GetArticle**, **GetArticleBody**, or **GetArticleHeader** methods.

See also

Attachments property
Body property
GetArticle method
GetArticleBody method
GetArticleHeader method
Header property
HeaderRecord property

GroupList property

See also Example

Applies to
TNMNNTP component

Declaration
```
property GroupList: TStringList;
```

Description
The **GroupList** property contains a list of the newsgroups on the current host.

Scope: Public
Accessability: Runtime, Read-Only

Note:
This property is set after a successful call to the **GetGroupList** method.

See also

GetGroupList method

Header property

See also Example

Applies to

TNMNNTP component

Declaration

```
property Header: TExStringList;
```

Description

The **Header** property contains the header for the current article retrieved from the news host.

Scope: Public
Accessability: Runtime, Read-Only

Note:
The Header property is set after a successful call to either the **GetArticle** method, the **GetArticleHeader** method, or the **GetArticleList** method.

See also

GetArticle method
GetArticleList method
GetArticleHeader method

HeaderRecord property

See also Example

Applies to
TNMNNTP component

Declaration
```
property HeaderRecord: TPostRecordType;
```

Description
The **HeaderRecord** property contains information contained in the article header, including the Subject, FromAddress of the sender, Reply-To address of the sender, and the time/date the article was posted.

Scope: Public
Accessability: Runtime Only

See also

TPostRecord type

HiMessage property

See also Example

Applies to
TNMNNTP component

Declaration
```
property HiMessage: integer;
```

Description
The **HiMessage** property specifies the highest message index (number) that is available in the current newsgroup.

Scope: Public
Accessability: Run-time, Read-only

See also

LoMessage property

LoMessage property

See also Example

Applies to
TNMNNTP component

Declaration
```
property LoMessage: integer;
```

Description
The **LoMessage** property specifies the lowest message index (number) that is available in the current newsgroup.

Scope: Public
Accessability: Run-time, Read-only

See also

HiMessage property

NewsDir property

See also

Applies to

TNMNNTP component

Declaration

```
property NewsDir: string;
```

Description

The **NewsDir** property specifies the directory name where cached news articles and headers will be stored.

NOTE
As of the current version of FastNet, the caching features of the TNMNNTP component are disabled.

See also

CacheMode property

ParseAttachments property

See also Example

Applies to
TNMNNTP component

Declaration
```
property ParseAttachments: boolean;
```

Description
The **ParseAttachments** property determines whether files attached to news messages will be decoded and stored on disk or not.

If the value is **TRUE**, files are parsed out from the body of the message, decoded, and stored on disk.
If the value is **FALSE**, files are left in the body of the news message.

Scope: Published
Accessability: Runtime, Designtime

See also

AttachFilePath property
Attachments property

Password property

See also Example

Applies to

TNMNNTP component

Declaration

```
property Password: string;
```

Description

The **Password** property is used when authentication is required to log on to a remote news host.

Scope: Published
Accessability: Runtime, Designtime

Notes:
The **UserID** property must be set to correspond with the **Password** property. If the password supplied is invalid, the **OnAuthenticationFailed** event is called. If a password is not supplied, and one is required, the **OnAuthenticationNeeded** event is called.

See also

OnAuthenticationFailed event
OnAuthenticationNeeded event
UserID property

Example

To recreate this example, you will need to create a new Delphi application.

Place 5 TEdits, 4 TButtons, 2 TListBoxes, a TMemo, a TOpenDialog, a TStatusBar, and a TNMNNTP.on the form.

Component Descriptions:
Edit1: User ID (if authentication is used)
Edit2: Password (if authentication is used)
Edit3: news host or IP address
Edit4: Address of the poster of the article
Edit5: Subject line for the article
Memo1: Body of the article
ListBox1: Newsgroup list
ListBox2: File attachment list

Insert the following code into Button1's OnClick event:

```
procedure TForm1.Button1Click(Sender: TObject);
begin
  with NMNNTP1 do
  begin
    if Connected then
      Disconnect
    else
      begin
        if Edit1.Text <> '' then
          begin
            UserID := Edit1.Text;
            Password := Edit2.Text;
          end;
        Host := Edit3.Text;
        Connect;
      end;
  end;
end;
```

When Button1 is clicked, if the **Connected** property is TRUE, the current connection is disconnected by calling the **Disconnect** method. If no connection is present, if Edit1 contains text. it is used as the **UserID** property, and the text in Edit2 is used for the **Password** property. The **Host** property is set to the value of the text in Edit3, and the **Connect** method is called.

Insert the following code into NMNNTP1's OnConnect event:

```
procedure TForm1.NMNNTP1Connect(Sender: TObject);
begin
  StatusBar1.SimpleText := 'Connected';
  NMNNTP1.GetGroupList;
end;
```

When a connection is established, the **OnConnect** event is called. The StatusBar is updated to inform the user that a connection has been established. The **GetGroupList** method is then called to retrieve a list of newsgroups on the news host the client is connected to.

Insert the following code into ListBox1's OnClick event:

```
procedure TForm1.ListBox1Click(Sender: TObject);
var
  S: String;
begin
  if ListBox1.ItemIndex <> -1 then
    begin
      S := ListBox1.Items[ListBox1.ItemIndex];
      NMNNTP1.SetGroup(S);
    end;
end;
```

ListBox1 is used for displaying the newsgroups available on the news host. When the mouse is clicked on ListBox1, the selected item (group) in the list is passed to the **SetGroup** method to set the current newsgroup to the group selected.

Insert the following code into NMNNTP1's OnGroupSelect event:

```
procedure TForm1.NMNNTP1GroupSelect(Sender: TObject);
begin
  Caption := 'Posting - '+NMNNTP1.SelectedGroup;
  if NMNNTP1.Posting then
    Caption := Caption + '(Posting allowed)'
  else
    Caption := Caption + '(Posting prohibited)';
end;
```

When a newsgroup has been selected, the **OnGroupSelect** event is called. Here, the caption for the application's form is set to display the **SelectedGroup** property, so the user will know the current newsgroup. The **Posting** property is also checked, and the caption is modified to display the ability to post in the current newsgroup.

Insert the following code into NMNNTP1's OnGroupListUpdate event:

```
procedure TForm1.NMNNTP1GroupListUpdate(name: String; FirstArticle,
  LastArticle: Integer; Posting: Boolean);
begin
  ListBox1.Items.Add(name);
end;
```

When a list of newsgroups is being retrieved, the **OnGroupListUpdate** event is called for each group listed. Here, the name of the newsgroup is added to ListBox1.

Insert the following code into NMNNTP1's OnDisconnect event:

```
procedure TForm1.NMNNTP1Disconnect(Sender: TObject);
begin
  StatusBar1.SimpleText := 'Disconnected';
end;
```

When a connection is closed with the remote host, the **OnDisconnect** event is called. Here, the status bar is updated to display the disconnected status.

Insert the following code into Button2's OnClick event:

```
procedure TForm1.Button2Click(Sender: TObject);
begin
  if OpenDialog1.Execute then
    ListBox2.Items.Add(OpenDialog1.FileName);
end;
```

When Button2 is clicked, the open dialog is displayed. If the Open button is clicked, the name of the file selected is added to ListBox2.

Insert the following code into Button4's OnClick event:

```
procedure TForm1.Button4Click(Sender: TObject);
begin
  NMNNTP1.PostAttachments.Text := ListBox2.Items.Text;
  NMNNTP1.PostBody.Text := Memo1.Text;
  with NMNNTP1.PostRecord do
  begin
    PrFromAddress := Edit4.Text;
    PrReplyTo := Edit4.Text;
    PrSubject := Edit5.Text;
    PrAppName := 'Netmasters Example News Posting';
    PrNewsGroups := NMNNTP1.SelectedGroup;
  end;
  NMNNTP1.PostHeader.Values['Organization'] := 'Our Organization';
  NMNNTP1.PostArticle;
end;
```

When Button4 is clicked, the value of the **PostAttachments** property is set to the list of filenames (if any) that are displayed in ListBox2. The **PostBody** property is set to the text that was entered into Memo1, and the **PostRecord** property is filled. The **PostHeader** property has a value called Organization added to it, specifying the organization of the poster, and the **PostArticle** method is called to post the news article on the remote news host.

Insert the following code into NMNNTP1's OnPosted event:

```
procedure TForm1.NMNNTP1Posted(Sender: TObject);
begin
  StatusBar1.SimpleText := 'Article Posted Successfully';
end;
```

When a news article has been successfully posted on the remote news host, the **OnPosted** event is called. Here, the statusbar is updated to inform the user that the article was posted successfully.

Insert the following code into NMNNTP1's OnPostFailed event:

```
procedure TForm1.NMNNTP1PostFailed(Sender: TComponent; Errno: Word;
  Errmsg: String);
```

```
begin
  ShowMessage('Posting Failed: '+Errmsg+' Error number '+IntToStr(Errno));
end;
```

When an article fails to be posted on the remote news host by the **PostArticle** method, the **OnPostFailed** event is called. Here, the ShowMessage procedure is used to display the error in posting to the user.

Insert the following code into NMNNTP1's OnGroupSelectRequired event:

```
procedure TForm1.NMNNTP1GroupSelectRequired(var Handled: Boolean);
var
  S: String;
begin
  if InputQuery('Group selection required', 'Enter the name of a group:', S) then
    begin
      NMNNTP1.SetGroup(S);
      Handled := TRUE;
    end;
end;
```

If a method is called (such as the **PostArticle** method) that requires a newsgroup to be previously selected with the **SetGroup** method, the **OnGroupSelectRequired** event is called. Here, the InputQuery function is used to obtain the name of a newsgroup. If the Ok button is clicked, the **SetGroup** method is called to set the group, and the Handled parameter is set to TRUE. If the Handled parameter were not modified, or set to FALSE (if the Ok button was not clicked), an exception would be raised.

Insert the following code into NMNNTP1's OnAuthenticationFailed event:

```
procedure TForm1.NMNNTP1AuthenticationFailed(Sender: TObject);
begin
  ShowMessage('Authentication failed');
end;
```

If the news host being connected to requires authentication (logging on with a user id and password), and the supplied User ID and/or Password are invalid, the **OnAuthenticationFailed** event is called. In this case, the ShowMessage procedure displays that the authentication failed.

Insert the following code into NMNNTP1's OnAuthenticationNeeded event:

```
procedure TForm1.NMNNTP1AuthenticationNeeded(var Handled: Boolean);
var
  AnID,
  APass: String;
begin
  if NMNNTP1.UserID = '' then
    begin
      if InputQuery('Authentication Required', 'Enter User ID', AnID) then
        begin
          NMNNTP1.UserId := AnID;
          Handled := TRUE;
        end;
    end;
```

```
  if NMNNTP1.Password = '' then
    begin
      if InputQuery('Authentication Required', 'Enter Password', APass) then
        begin
          NMNNTP1.Password := APass;
          Handled := TRUE;
        end;
    end;
end;
```

If authentication is required by the news host being connected to, and either the **UserID** property or the **Password** property are empty, the **OnAuthenticationRequired** event is called. Here, the InputQuery function is used to obtain the missing information. If the information is not suppled, an exception is raised. Otherwise, the Handled parameter is set to TRUE, and the authentication is attempted.

PostAttachments property

See also Example

Applies to
TNMNNTP component

Declaration
```
property PostAttachments: TStringList;
```

Description
The **PostAttachments** property is used for storing the names of files that are to be attached to the outgoing news message.

Scope: Published
Accessability: Runtime, Designtime

Notes:
The files to be attached to the outgoing news article should be put into the stringlist one filename per line.

See also

PostArticle method
PostBody property
PostHeader property
PostRecord property

PostBody property

See also Example

Applies to
TNMNNTP component

Declaration
```
property PostBody: TExStringList;
```

Description
The **PostBody** property specifies the body of the news article to post.

Scope: Published
Accessability: Runtime, Designtime

See also

PostArticle method
PostAttachments property
PostHeader property
PostRecord property

PostHeader property

See also Example

Applies to
TNMNNTP component

Declaration
```
property PostHeader: TExStringList;
```

Description
The **PostHeader** property specifies the header for a news article to be posted on the remote NNTP host.

Scope: Published
Accessability: Runtime, Designtime

Notes:
Custom header fields can be added to the outgoing article's header by accessing the **Values** property of **PostHeader**. See the example for more details.

See also

PostArticle method
PostAttachments property
PostBody property
PostRecord property

Posting property

See also Example

Applies to
TNMNNTP component

Declaration
```
property Posting: boolean;
```

Description
The **Posting** property specifies whether articles may be posted in the current newsgroup.

Scope: Public
Accessability: Runtime, Read-Only

See also

PostArticle method

PostRecord property

See also Example

Applies to
TNMNNTP component

Declaration
```
property PostRecord: TPostRecordType;
```

Description
The PostRecord property specifies the most common header fields used when posting a news article on an NNTP host.

Scope: Published
Accessability: Runtime, Designtime

See also

PostArticle method
PostAttachments property
PostBody property
PostHeader property

SelectedGroup property

See also Example

Applies to
TNMNNTP component

Declaration
```
property SelectedGroup: string;
```

Description
The **SelectedGroup** property specifies the name of the currently selected newsgroup.

Scope: Public
Accessability: Runtime, Read-Only

Notes:
The **SelectedGroup** property changes when the **SetGroup** method is called.

See also

SetGroup method

UserId property

See also Example

Applies to
TNMNNTP component

Declaration
```
property UserId: string;
```

Description
The **UserID** property specifies a user ID if authentication is necessary on the remote host.

Scope: Published
Accessability: Runtime, Designtime

Notes:
The **Password** property must be set to correspond with the **UserID** property. If the User ID supplied is invalid, the **OnAuthenticationFailed** event is called. If a UserID is not supplied, and one is required, the **OnAuthenticationNeeded** event is called.

See also

OnAuthenticationFailed event
OnAuthenticationNeeded event
Password property

GetArticle method

See also Example

Applies to
TNMNNTP component

Declaration
```
procedure GetArticle(Ref: integer);
```

Description
The **GetArticle** method retrieves the article specified from the remote NNTP host.

Parameters:
The **Ref** parameter specifies the index of the message to retrieve from the NNTP host.

Notes:
If the **Ref** parameter is not a valid article number, the **OnInvalidArticle** event is called.
When the article has been successfully gotten, the **OnArticle** event is called. If caching has been enabled, the **OnArticleCacheUpdate** event will be called. The **Header**, **Body**, and **HeaderRecord** properties are updated to contain the retrieved article after this method has been called successfully. If there are any file attachments, they are listed in the **Attachments** property.

See also

OnInvalidArticle event
OnArticle event
OnArticleCacheUpdate event
Header property
Body property
HeaderRecord property
Attachments property

GetArticleBody method

See also Example

Applies to

TNMNNTP component

Declaration

```
procedure GetArticleBody(Ref: integer);
```

Description

The GetArticleBody method retrieves the body of the article specified.

Parameters:

The **Ref** parameter specifies the index of the article to retrieve the body of.

Notes:

When the body has been retrieved from the remote NNTP host, the **OnBody** event is called.
Not all servers support this method.

See also

GetArticle method
GetArticleHeader method

GetArticleHeader method

See also Example

Applies to

TNMNNTP component

Declaration

```
procedure GetArticleHeader(Ref: integer);
```

Description

The **GetArticleHeader** method retrieves the header of the article specified.

Parameters:

The **Ref** parameter specifies the index of the article to retrieve the header of.

Notes:

When the body has been retrieved from the remote NNTP host, the **OnHeader** event is called.

See also

GetArticle method
GetArticleBody method

GetArticleList method

See also Example

Applies to
TNMNNTP component

Declaration
```
procedure GetArticleList(All: boolean; ArticleNumber: integer);
```

Description
The **GetArticleList** method retrieves a list of articles in the current newsgroup.

Parameters:
The **All** parameter specifies whether you want to retrieve a list of all the articles in the newsgroup, or only some of them. Pass **TRUE** if you wish to retrieve a list of all articles, and **FALSE** if you wish only to retrieve a partial list.
The **ArticleNumber** parameter specifies the number of the article to begin the listing at.

Notes:
The **OnHeaderList** event is called for each article listed.

See also

OnHeaderList event

GetGroupList method

See also Example

Applies to
TNMNNTP component

Declaration
```
procedure GetGroupList;
```

Description
The **GetGroupList** method retrieves a list of available newsgroups from the remote host.

Notes:
The **OnGroupListUpdate** event is called for each newsgroup listed.

See also

OnGroupListUpdate event

PostArticle method

See also Example

Applies to

TNMNNTP component

Declaration

```
procedure PostArticle;
```

Description

The **PostArticle** method posts a news article in the currently selected newsgroup.

Notes:

The article posted by the PostArticle method is defined by the **PostRecord** property, the **PostHeader** property, the **PostBody** property, and the **PostAttachments** property. If the message is posted successfully, the **OnPosted** event is called. If the message fails to post, the **OnPostFailed** event is called.

See also

PostRecord property
PostHeader property
PostBody property
PostAttachments property
OnPosted event
OnPostFailed event

SetGroup method

See also Example

Applies to

TNMNNTP component

Declaration

```
procedure SetGroup(Group: string);
```

Description

The **SetGroup** method sets the currently active newsgroup to the group specified.

Parameters:

The **Group** parameter specifies the newsgroup to set as the active group.

Notes:

If the specified group exists, the **OnGroupSelect** event is called. The **SelectedGroup** property will also change when the SetGroup method has been called successfully.

See also

OnGroupSelect event
SelectedGroup property

OnAbort event

See also Example

Applies to

TNMNNTP component

Declaration

```
property OnAbort: TNotifyEvent;
```

Description

The **OnAbort** event is called when the **Abort** method has been called.

See also

Abort method

OnArticle event

See also Example

Applies to
TNMNNTP component

Declaration
```
property OnArticle: TNotifyEvent;
```

Description
The **OnArticle** event is called when the **GetArticle** method is called successfully.

Notes:
The article retrieved can be accessed through the **Body** property, the **Header** property, and the **Attachments** property.

See also

Body property
Header property
Attachments property

OnArticleCacheUpdate event

See also

Applies to

TNMNNTP component

Declaration

```
property OnArticleCacheUpdate: THeaderCacheEvent;
```

Description

The **OnArticleCacheUpdate** event is called when an article has been received from the remote host, and caching is enabled. At this point in time, caching is not available even if the **CacheMode** property is set accordingly.

See also

OnBodyCacheUpdate event
OnGroupListCacheUpdate event
OnHeaderCacheUpdate event
OnHeaderListCacheUpdate event

OnAuthenticationFailed event

See also Example

Applies to

TNMNNTP component

Declaration

```
property OnAuthenticationFailed: TNotifyEvent;
```

Description

The **OnAuthenticationFailed** event is called when the user ID and password supplied to connect to a secure are invalid.

See also

OnAuthenticationNeeded event

OnAuthenticationNeeded event

See also Example

Applies to

TNMNNTP component

Declaration

```
property OnAuthenticationNeeded: THandlerEvent;
```

Description

The **OnAuthenticationNeeded** event is called when either the **UserID** property or the **Password** property are blank, and authentication is needed to log on to the remote news host.

See also

OnAuthenticationFailed event

OnBody event

See also Example

Applies to

TNMNNTP component

Declaration

```
property OnBody: TNotifyEvent;
```

Description

The **OnBody** event is called when the **GetArticleBody** method is executed successfully.

See also

OnBodyCacheUpdate event

OnBodyCacheUpdate event

See also

Applies to

TNMNNTP component

Declaration

```
property OnBodyCacheUpdate: THandlerEvent;
```

Description

The **OnBodyCacheUpdate** event is called when the **GetArticleBody** method is executed successfully, and caching is enabled.

Notes:

Caching is disabled regardless of the value of the **CacheMode** property at this time.

See also

OnArticleCacheUpdate event
OnGroupListCacheUpdate event
OnHeaderCacheUpdate event
OnHeaderListCacheUpdate event

OnGroupListCacheUpdate event

See also

Applies to

TNMNNTP component

Declaration

```
property OnGroupListCacheUpdate: TGroupRetrievedCacheEvent;
```

Description

The **OnGroupListCacheUpdate** event is called when the **GetGroupList** method has executed successfully, and caching is enabled. This event is called once for each group that is listed.

Notes:

Caching is disabled regardless of the value of the **CacheMode** property at this time.

See also

OnArticleCacheUpdate event
OnBodyCacheUpdate event
OnHeaderCacheUpdate event
OnHeaderListCacheUpdate event

OnGroupListUpdate event

See also Example

Applies to

TNMNNTP component

Declaration

```
property OnGroupListUpdate: TGroupRetrievedEvent;
```

Description

The **OnGroupListUpdate** event is called when the **GetGroupList** method has been excuted successfully.

This event is called once for each group that is listed.

See also

GetGroupList method

OnGroupSelect event

See also Example

Applies to

TNMNNTP component

Declaration

```
property OnGroupSelect: TNotifyEvent;
```

Description

The **OnGroupSelect** event is called when the **SetGroup** method is executed successfully.

See also

SetGroup method

OnGroupSelectRequired event

See also Example

Applies to

TNMNNTP component

Declaration

```
property OnGroupSelectRequired: THandlerEvent;
```

Description

The **OnGroupSelectRequired** event is called when a method that requires a group to be selected is called.

Notes:

The **GetArticle** method, **GetArticleBody** method, **GetArticleHeader** method, **GetArticleList** method, and the **PostArticle** method all require a newsgroup to be selected with the **SetGroup** method before they are used.

See also

GetArticle method
GetArticleBody method
GetArticleHeader method
GetArticleList method
PostArticle method
SetGroup method

OnHeader event

See also Example

Applies to

TNMNNTP component

Declaration

```
property OnHeader: TNotifyEvent;
```

Description

The **OnHeader** event is called when the header of a news article has been retrieved with the **GetArticleHeader** method.

See also

GetArticleHeader method
Header property

OnHeaderCacheUpdate event

See also

Applies to

TNMNNTP component

Declaration

```
property OnHeaderCacheUpdate: THeaderCacheEvent;
```

Description

The **OnHeaderCacheUpdate** event is called when the header of a news article has been retrieved with the **GetArticleHeader** method and caching is enabled.

Note:
Caching does not function regardless of the **CacheMode** property at this time.

See also

GetArticleHeader method
OnArticleCacheUpdate event
OnBodyCacheUpdate event
OnGroupListCacheUpdate event
OnHeaderListCacheUpdate event

OnHeaderList event

See also Example

Applies to

TNMNNTP component

Declaration

```
property OnHeaderList: TNotifyEvent;
```

Description

The **OnHeaderList** event is called for each article listed when the **GetArticleList** method has been executed successfully.

Notes:

When this event is called, the values of the **Header** property and the **HeaderRecord** property should be read, because they are changed at each call to this event.

See also

GetArticleList method
Header property
HeaderRecord property

OnHeaderListCacheUpdate event

See also

Applies to

TNMNNTP component

Declaration

```
property OnHeaderListCacheUpdate: THandlerEvent;
```

Description

The **OnHeaderListCacheUpdate** event is called for each article listed when the **GetArticleList** method has been executed successfully, and caching is enabled.

Notes:

This event is never called regardless of the value of the **CacheMode** property at this time.

See also

CacheMode property
GetArticleHeader method
OnArticleCacheUpdate event
OnBodyCacheUpdate event
OnGroupListCacheUpdate event
OnHeaderCacheUpdate event

OnInvalidArticle event

See also Example

Applies to

TNMNNTP component

Declaration

```
property OnInvalidArticle: TNotifyEvent;
```

Description

The **OnInvalidArticle** event is called when an article ID number passed to the **GetArticle**, **GetArticleBody**, or **GetArticleHeader** method does not exist or is invalid.

See also

GetArticle method
GetArticleBody method
GetArticleHeader method

OnPosted event

See also Example

Applies to

TNMNNTP component

Declaration

```
property OnPosted: TNotifyEvent;
```

Description

The **OnPosted** event is called when the **PostArticle** method has successfully posted an article on the remote news host.

Notes:

If the posting of the article fails, the **OnPostFailed** event is called.

See also

OnPostFailed event
PostArticle method

OnPostFailed event

See also Example

Applies to

TNMNNTP component

Declaration

```
property OnPostFailed: TOnErrorEvent;
```

Description

The **OnPostFailed** event is called when the **PostArticle** method fails to post an article on the remote news host.

Notes:

If the posting is successful, the **OnPosted** event is called.

See also

OnPosted event
PostArticle method

TCacheMode type

Unit
NMNNTP

Declaration
```
type
  TCacheMode = (cmMixed, cmRemote, cmLocal);
```

Description
The TCacheMode type is used for determining the style of news article caching that will be used.

TGroupRetrievedCacheEvent type

Unit
NMNNTP

Declaration
```
type
  TGroupRetrievedCacheEvent = procedure(var Handled: boolean; name: string;
FirstArticle, LastArticle: integer; Posting: boolean) of object;
```

Description
The TGroupRetrievedCacheEvent type is used for the **OnGrouplistCacheUpdate** event.

The **Handled** parameter specifies whether the default caching action of the component will be carried out.
The **name** parameter specifies the name of the newsgroup being listed.
The **FirstArticle** parameter specifies the article number of the first article in the newsgroup. This is **NOT** always 1.
The **LastArticle** parameter specifies the article number of the last article in the newsgroup.
The **Posting** parameter specifies whether posting articles is allowed in this newsgroup.

TGroupRetrievedEvent type

Unit
NMNNTP

Declaration
```
type
  TGroupRetrievedEvent = procedure(name: string; FirstArticle, LastArticle:
integer; Posting: boolean) of object;
```

Description
The TGroupRetrievedEvent type is used for the **OnGrouplistUpdate** event.

The **name** parameter specifies the name of the newsgroup being listed.
The **FirstArticle** parameter specifies the article number of the first article in the newsgroup. This is **NOT** always 1.
The **LastArticle** parameter specifies the article number of the last article in the newsgroup.
The **Posting** parameter specifies whether posting articles is allowed in this newsgroup.

THeaderCacheEvent type

Unit
NMNNTP

Declaration
```
type
  THeaderCacheEvent = procedure(var Handled: boolean; IdNo: integer; From,
Subject, MsgId, Date: string; ArticleNo: integer) of object;
```

Description
The THeaderCacheEvent type is used for the **OnHeaderCacheUpdate** event

The **Handled** parameter specifies whether the component's default caching will take place.
The **IdNo** parameter specifies the article ID number of the header being retrieved.
The **From** parameter specifies the person that posted the article on the news host.
The **Subject** parameter specifies the subject line of the article
The **MsgId** parameter specifies the message ID of the article.
The **Date** parameter specifies the date the article was posted on the news host.
The **ArticleNo** parameter specifies the article number of the article

Legend

- Run-time only
- Read-Only
- Published
- Protected
- Key item

Heirarchy

TObject
|
TPersistent
|
TComponent
|
TPowersock