

NMSTRM unit

The NMSTRM unit contains the TNMStrm component for sending streams across the internet or an intranet, and the TNMStrmServ component, which is used for receiving streams sent by the TNMMsg component.

Purpose of the FastNet Stream Components

The main purpose of the components in this unit is to provide an example for you, the user, so that you may use the base classes provided with the FastNet Tools for Delphi to solve custom internet/intranet development problems with custom internet protocols. You may find the TNMStrm and TNMStrmServ components useful to you, but by examining the source code, you will discover how to use the base classes, TPowersock and TNMGeneralServer, to create your own components to more specifically address your needs.

Components

[TNMStrm](#)

[TNMStrmServ](#)

Types

[TStrmEvent](#)

TNMStrm component

[Heirarchy](#)

[Properties](#)

[Methods](#)

[Events](#)

[Tasks](#)

Unit

[NMSTRM](#)

Description

The TNMStrm component is used for sending streams to a stream server across the internet or an intranet.

TNMStrm Properties

[TNMStrm](#)

[Legend](#)

In TNMStrm

[FromName](#)

Derived from TPowersock

- [About](#)
- ▶ [BeenCanceled](#)
 - ▶
- ▶ [BeenTimedOut](#)
 - ▶
- ▶ [BytesRecvd](#)
 - ▶
- ▶ [BytesSent](#)
 - ▶
- ▶ [BytesTotal](#)
- ▶ [Connected](#)
 - ▶
- ▶ [Handle](#)
- [Host](#)
 - ▶ [LastErrorNo](#)
 - ▶
- ▶ [LocalIP](#)
- [Port](#)
 - [Proxy](#)
 - [ProxyPort](#)
 - ▶
- ▶ [RemotelIP](#)
 - ▶
- ▶ [ReplyNumber](#)
- [ReportLevel](#)
- ▶ [Status](#)
 - [TimeOut](#)
 - ▶
- ▶ [TransactionReply](#)
 - ▶
- ▶ [WSAInfo](#)

Derived from TComponent

- ▶ [ComObject](#)
- ▶ [ComponentCount](#)
- [ComponentIndex](#)
- ▶ [Components](#)
 - ▶ [ComponentState](#)
 - ▶ [ComponentStyle](#)
 - [DesignInfo](#)
 - ▶ [Owner](#)
 - [Tag](#)

VCLComObject

TNMStrm Methods

[TNMStrm](#)


[Legend](#)

In TNMStrm


 [PostIt](#)


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](#)

[ClassNames](#)

[ClassParent](#)

[ClassType](#)

[CleanupInstance](#)

[DefaultHandler](#)
[Dispatch](#)
[FieldAddress](#)
[Free](#)
[FreeInstance](#)
[GetInterface](#)
[GetInterfaceEntry](#)
[GetInterfaceTable](#)
[InheritsFrom](#)
[InitInstance](#)
[InstanceSize](#)
[MethodAddress](#)
[MethodName](#)
[NewInstance](#)

TNMStrm Events

[TNMStrm](#)

[Legend](#)

In TNMStrm

[OnMessageSent](#)

Derived from TPowersock

■ [OnAccept](#)

🔑 [OnConnect](#)

[OnConnectionFailed](#)

■ [OnConnectionRequired](#)

🔑 [OnDisconnect](#)

■ [OnError](#)

[OnHostResolved](#)

[OnInvalidHost](#)

[OnPacketRecvd](#)

[OnPacketSent](#)

[OnRead](#)

[OnStatus](#)

About the TNMStrm component

[TNMStrm reference](#)

Purpose

The purpose of the TNMStrm component is to send streams to a stream server across the internet or an intranet. It must be paired with the TNMStrmServ component to provide 2-way communication. The use of this component requires a 32-bit Winsock stack, WSOCK32.DLL, which is available from many vendors, and is included with Windows 95.

Tasks

Before you can send a stream, you must first set the **Host** property to the remote host you wish to send the stream to. You must also set the **Port** property to correspond with the port of the remote stream host. Set the **FromName** property so that the recipient of the stream will know who it is coming from. Once these properties are set, you can send streams by calling the **PostIt** method.

FromName property

[Example](#)

Applies to

[TNMStrm](#) component

Declaration

```
property FromName: string;
```

Description

The **FromName** property specifies the name of the sender of outgoing streams. This property is picked up by the server so recipients of data will know who it is coming from.

Scope: Published

Accessibility: Runtime, designtime

Example

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

Place a TMemo, 2 TEdits, a TButton, a TNMStrm, and a TNMStrmServ on the form.

Memo1: Status/Message window
Edit1: Name of sender
Edit2: Text to send as a stream
Button1: Sends the stream

Insert the following code into Button1's OnClick event:

```
procedure TForm1.Button1Click(Sender: TObject);  
var  
    MS: TMemoryStream;  
begin  
    NMStrm1.Host := '127.0.0.1';  
    NMStrm1.FromName := Edit1.Text;  
    MS := TMemoryStream.Create;  
    try  
        StreamLn(MS, Edit2.Text);  
        NMStrm1.PostIt(MS);  
    finally  
        MS.Free;  
    end;  
end;
```

When Button1 is clicked, the **Host** property of NMStrm1 is set to 127.0.0.1 (local host) to communicate with the NMStrmServ on the form. The **FromName** property of NMStrm1 is set to the text in Edit1. a Memory Stream is created, and the **StreamLn** procedure from the PSock unit is used to copy the text from Edit2 into the stream. The **PostIt** method of NMStrm1 is used to send the stream to the remote host. Note that the Memory Stream MS is used within a **try...finally** block so that its resources will be freed.

Insert the following code into NMStrm1's OnMessageSent event:

```
procedure TForm1.NMStrm1MessageSent(Sender: TObject);  
begin  
    Memo1.Lines.Add('Stream Sent');  
end;
```

When a message has been sent with the **PostIt** method, the **OnMessageSent** event is called. In this case, the event adds a line to Memo1 to inform the user that the stream was sent.

Insert the following code into NMStrmServ1's OnMSG event:

```
procedure TForm1.NMStrmServ1MSG(Sender: TComponent; const sFrom: String;  
    strm: TStream);  
var  
    SS: TStringStream;  
begin  
    SS := TStringStream.Create("");  
    try
```

```
    SS.CopyFrom(strm, strm.size);
    Memo1.Lines.Add(sFrom+' sent '+SS.DataString);
finally
    SS.Free;
end;
end;
```

When there is an incoming stream, NMStrmServ1's **OnMSG** event is called. a TStringStream, SS, is created, and the data from the incoming stream **strm** is copied into it. Memo1 is then updated to display the name of the sender and the data send by adding the **sFrom** parameter and the DataString property of the TStringStream. Note that SS is used within a **try...finally** block so that its resources will be freed.

PostIt method

[See also](#) [Example](#)

Applies to

[TNMStrm](#) component

Declaration

```
function PostIt(const sStrm: TStream): string;
```

Description

The PostIt method sends the stream passed as the **sStrm** parameter to the remote host specified by the **Host** property.

Parameters:

The **sStrm** parameter is any TStream (or descendant) that contains data to be sent to the remote host.

Return Value:

The return value of this function is an OK response from the server if the stream was received properly.

Note:

The **PostIt** method handles all connecting/disconnecting with the remote host.

The **OnMessageSent** event is called when the message has been sent.

See Also

[OnMessageSent](#) event

OnMessageSent event

[See also](#) [Example](#)

Applies to

[TNMStrm](#) component

Declaration

```
property OnMessageSent: TNotifyEvent;
```

Description

The **OnMessageSent** event is called when the **PostIt** method has executed successfully.

See also

[PostIt](#) method



TNMStrmServ component

[Heirarchy](#)

[Properties](#)

[Methods](#)

[Events](#)

[Tasks](#)

Unit

[NMSTRM](#)

Description





The TNMStrmServ component is used to received streams sent with the TNMStrm component.

TNMStrmServ Properties

[TNMStrmServ](#)

[Legend](#)

Derived from TPowersock

- [About](#)
- ▶ [BeenCanceled](#)
 - ▶
- ▶ [BeenTimedOut](#)
- ▶ [BytesRecvd](#)
 - ▶
- ▶ [BytesSent](#)
 - ▶
- ▶ [BytesTotal](#)
- ▶ 
- ▶ [Connected](#)
 - ▶
- ▶ [Handle](#)
 - ▶ 
- [Host](#)
 - ▶ [LastErrorNo](#)
 - ▶
- ▶ [LocalIP](#)
 - ▶ 
- [Port](#)
 - [Proxy](#)
 - [ProxyPort](#)
 - ▶
- ▶ [RemotelIP](#)
 - ▶
- ▶ [ReplyNumber](#)
- [ReportLevel](#)
- ▶ 
- ▶ [Status](#)
 - [TimeOut](#)
 - ▶
- ▶ [TransactionReply](#)
 - ▶
- ▶ [WSAInfo](#)

Derived from TComponent

- ▶ [ComObject](#)
- ▶ [ComponentCount](#)
- [ComponentIndex](#)
- ▶ [Components](#)
 - ▶ [ComponentState](#)
 - ▶ [ComponentStyle](#)
 - [DesignInfo](#)
 - ▶ [Owner](#)
 - [Tag](#)
 - [VCLComObject](#)

TNMStrmServ Properties

[TNMStrmServ](#)

[Legend](#)

Derived from TNMGeneralServer

 [Serve](#)

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](#)

[ClassNames](#)

[ClassParent](#)

[ClassType](#)

[CleanupInstance](#)

[DefaultHandler](#)
[Dispatch](#)
[FieldAddress](#)
[Free](#)
[FreeInstance](#)
[GetInterface](#)
[GetInterfaceEntry](#)
[GetInterfaceTable](#)
[InheritsFrom](#)
[InitInstance](#)
[InstanceSize](#)
[MethodAddress](#)
[MethodName](#)
[NewInstance](#)

TNMStrmServ Events

[TNMStrmServ](#)

[Legend](#)

In TNMStrmServ

▶ [OnMSG](#)

Derived from TNMGeneralServer

▶ [OnClientContact](#)

Derived from TPowersock

— [OnConnect](#)

[OnConnectionFailed](#)

[OnDisconnect](#)

[OnHostResolved](#)

[OnInvalidHost](#)

[OnPacketRecvd](#)

[OnPacketSent](#)

[OnRead](#)

[OnStatus](#)

About the TNMStrmServ component

[TNMStrmServ reference](#)

Purpose

The purpose of the TNMStrmServ component is to receive streams sent with the TNMStrm component. The use of this component requires a 32-bit Winsock stack, WSOCK32.DLL, which is available from many vendors, and is included with Windows 95.

Tasks

While you are designing your application, set the **Port** property in the object inspector to the port that your message server will listen on, if you wish it to differ from the default.

To receive and process messages sent to the message server, write an event handler for the **OnMSG** event.

OnMSG event

[Example](#)

Applies to

[TNMStrmServ](#) component

Declaration

property OnMSG: [TStrmEvent](#);

Description

The **OnMSG** event is called when there is an incoming stream present.

Event Parameters:

The **sFrom** parameter specifies the name of the person who sent the stream

The **strm** parameter is the stream that was sent.

Notes:

You must read the data out of the **strm** parameter before the exit of the event, or the data will be lost.

TStrmEvent type

Unit

[NMSTRM](#)

Declaration

```
type TStrmEvent = procedure (Sender: TComponent; const sFrom: string; strm:  
Tstream) of object;
```

Description

The TStrmEvent event type is used for capturing incoming streams.

The sFrom parameter specifies the name of the sender of the stream.
The strm parameter is the stream sent by the user.

Legend

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

Heirarchy

TObject

|

TPersistent

|

TComponent

|

TPowersock

Heirarchy

TObject

|

TPersistent

|

TComponent

|

TPowersock

|

TNMGeneralServer

