



## TNMTIME component

[Heirarchy](#)

[Properties](#)

[Methods](#)

[Events](#)

[Tasks](#)

### Unit

NMTime

### Description

The TNMTIME component is used for getting the time from Internet time servers, as described in RFC 868.

## TNMTime Properties

[TNMTime](#)

[Legend](#)

### In TNMTime

- ▶
- ▶ [TimeInt](#)
- ▶
- ▶ [TimeStr](#)

### 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

## TNMTime Methods

[TNMTime](#)

[Legend](#)

### 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

## TNMTime Events

[TNMTime](#)

[Legend](#)

### Derived from TPowersock

- [OnAccept](#)
- [OnConnect](#)
- [OnConnectionFailed](#)
- [OnConnectionRequired](#)
- [OnDisconnect](#)
- [OnError](#)
- [OnHostResolved](#)
- [OnInvalidHost](#)
- [OnPacketRecv](#)
- [OnPacketSent](#)
- [OnRead](#)
- [OnStatus](#)

## About the TNMTime component

[TNMTime reference](#)

### Purpose

The purpose of the TNMTime component is to get the time from an internet/intranet time server. This component requires a 32 bit TCP/IP stack, WSOCK32.DLL, which can be obtained from many vendors, and is included with Windows 95.

### Tasks

Before you can get the time from an internet time server, you must first know the host name or dotted IP address of the daytime host, and assign that value to the **Host** property. Normally, the **Port** property need not be modified, unless the host is listening on a non-standard port.

Once the host is set, just read the value in the [TimeStr](#) property, and that will give you the time from the specified host.

# TimeInt property

[See also](#)

[Example](#)

## Applies to

[TNMTime](#) component

## Declaration

**property** TimeInt: longint;

## Description

The TimeInt property returns the number of seconds since Midnight, January 1, 1900. 1 would be 12:00:01am, January 1, 1900.

**Scope:** Public

**Accessability:** Runtime, Read-only

## Note:

The **Host** property must be set to a valid internet Time server before the value of the TimeInt property can be read.



## See also

[TimeStr](#) property

## Example

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

Place a TNetTime, 2 TEdit, and a TButton on the form.

Insert the following code into Button1's OnClick event:

```
void __fastcall TForm1::Button1Click(TObject *Sender)
{
    NNetTime1->Host = "tmc.edu";
    NNetTime1->Timeout = 5000;
    Edit1->Text = NNetTime1->TimeStr;
    Edit2->Text = IntToStr(NNetTime1->TimeInt);
}
```

When Button1 is clicked, the **Host** property is set to tmc.edu, which is a valid time server. The **Timeout** property is set to 5000 milliseconds (5 seconds), so that a timeout doesn't occur before the time is gotten. Edit1 is set to display the time by reading the **TimeStr** property, which handles all connecting and disconnecting with the server. Edit2 is set to display the **TimeInt** property, which also handles connecting/disconnecting with the time server.

# TimeStr property

[See also](#)

[Example](#)

## Applies to

[TNMTime](#) component

## Declaration

**property** TimeStr: **string**;

## Description

The TimeStr property returns the Date from the remote host as a string.

**Scope:** Public

**Accessibility:** Runtime, Readonly






## Notes:

The **Host** property must be set to a valid internet Time server before the value of the TimeStr property can be read.

## See also

[TimeInt](#) property

## Legend

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

# Heirarchy

TObject



TPersistent



TComponent



TPowersock

# TimeOut property

[See also](#)

[Example](#)

## Declaration

**property** TimeOut: Integer;

## Description

The TimeOut property specifies the amount of time (in milliseconds) to wait for a response from the socket before an exception is raised and the current operation is aborted. If TimeOut is 0, an exception is never raised, and operations never time out.

**Default:** 500

**Scope:** Published

**Accessibility:** Designtime, Runtime

## See also

TimeStr property



