



## TNMURL component

[Heirarchy](#)

[Properties](#)

[Methods](#)

[Events](#)

[Tasks](#)

### Description

The TNMURL component decodes URL data into a readable string, and encodes standard strings into URL data format.

## TNMURL Properties

[TNMURL](#)

[Legend](#)

### In TNMUDP



 [Decode](#)










 [Encode](#)



 [InputString](#)

### Derived from TComponent

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

## **TNMURL Methods**

[TNMURL](#)

[Legend](#)

### **Derived from TComponent**

[DestroyComponents](#)

[Destroying](#)

[FindComponent](#)

[FreeNotification](#)

[FreeOnRelease](#)

[GetParentComponent](#)

[HasParent](#)

[InsertComponent](#)

[RemoveComponent](#)

[SafeCallException](#)

### **Derived from TPersistent**

[Assign](#)

[GetNamePath](#)

### **Derived from TObject**

[ClassInfo](#)

[ClassName](#)

[ClassNamels](#)

[ClassParent](#)

[ClassType](#)

[CleanupInstance](#)

[DefaultHandler](#)

[Dispatch](#)

[FieldAddress](#)

[Free](#)

[FreeInstance](#)

[GetInterface](#)

[GetInterfaceEntry](#)

[GetInterfaceTable](#)

[InheritsFrom](#)

[InitInstance](#)

[InstanceSize](#)

[MethodAddress](#)

[MethodName](#)

[NewInstance](#)

## TNMURL Properties

TNMURL

Legend

In TNMUDP

OnError

# About the TNMURL component

[TNMURL reference](#)

## Purpose

The purpose of the TNMURL component is to encode strings into URL format for HTTP transmission, and to decode URL data into readable strings.

## Tasks

Using the TNMURL component is easy. Simply set the [InputString](#) property, and then the [Encode](#) property will contain the URL encoded string, and the [Decode](#) property will contain the decoded string. If an error occurs, then the [OnError](#) event will be called.

## Decode property

[See Also](#) — [Example](#)

### Applies to

[TNMURL](#) component

### Declaration

```
property Decode: String;
```

### Description

The Decode property contains the decoded string, based on the encoded string in the InputString property. If InputString is not an encoded URL, then Decode is the same as InputString.

**Scope:** Public

**Accessibility:** RunTime, ReadOnly

## See Also

[Encode](#) property

[InputString](#) property

[OnError](#) event

# Encode property

[See Also](#) — [Example](#)

## Applies to

[TNMURL](#) component

## Declaration

```
property Encode: String;
```

## Description

The Encode property contains the URL-encoded copy of the InputString property.

**Scope:** Public

**Accessibility:** RunTime, ReadOnly



## Example

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

Place 2 TEdits, 2 TButtons, and a TNMURL on the form.

Edit1: InputString  
Edit2: Component Output (encoded/decoded Edit1)  
Button1: Encode  
Button2: Decode

Insert the following code into Button1's OnClick event:

```
procedure TForm1.Button1Click(Sender: TObject);  
begin  
    // Encoding  
    NMURL1.InputString := Edit1.Text;  
    Edit2.Text := NMURL1.Encode;  
end;
```

When Button1 is clicked, the text in Edit1 is stored in the **InputString** property. When the **Encode** property is read, the data in Edit1 is encoded, and displayed in Edit2.

Insert the following code into Button2's OnClick event:

```
procedure TForm1.Button2Click(Sender: TObject);  
begin  
    NMURL1.InputString := Edit1.Text;  
    Edit2.Text := NMURL1.Decode;  
end;
```

When Button2 is clicked, the text in Edit1 is stored in the **InputString** property. When the **Decode** property is read, the data in Edit1 is decoded and displayed in Edit2.

Insert the following code into NMURL1's OnError event:

```
procedure TForm1.NMURL1Error(Sender: TObject; Operation, ErrMsg: String);  
begin  
    ShowMessage(Operation+' '+ErrMsg);  
end;
```

If an error occurs in the decoding or encoding of a string, the **OnError** event is called. In this instance, a message is displayed to the user stating that there was an error during the operation specified by the **Operation** parameter, and the error message is displayed by showing the **ErrMsg** parameter.

## See Also

[Decode](#) property

[InputString](#) property

[OnError](#) event

# InputString property

[See Also](#) — [Example](#)

## Applies to

[TNMURL](#) component

## Declaration

```
property InputString: String;
```

## Description

The InputString property contains either the encoded data to be decoded, or a standard string to be encoded.

**Scope:** Published

**Accessibility:** DesignTime, RunTime

## See Also

[Encode](#) property  
[Decode](#) property

## OnError event

[Example](#)

### Applies to

[TNMURL](#) component

### Declaration

**property** OnError: [TOnErrorEvent](#);

### Description

The OnError event is called when there is an error in the encoding or decoding of an URL.

### Event Parameters:

The Operation string contains the operation that caused the error.

The ErrMsg string contains a short description of the error encountered.

## TOnErrorEvent type

### Declaration

#### type

TOnErrorEvent = **procedure**(Sender: TObject; Operation, ErrMsg: **String**);

#### Description

The TOnErrorEvent specifies the type used by the OnError event of the TNMURL component.

The Operation parameter describes the operation the error occurred in.

The ErrMsg parameter describes the error.

# Heirarchy

TObject



TPersistent



TComponent

## Legend

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



