

## To Get Started

[TElasticForm](#) component:

### Version 2.06

Just drop the component on your form. The proportions of all your components will be maintained at run time during resizing. Your form also becomes independent of screen resolution or font size (small fonts, large fonts).

**Important:** Use true type fonts only to ensure proper resizing especially at small form sizes.

To see more details on the use of the component and additional uses review its [properties](#)

[Installation](#)

[To obtain a Registered Copy](#)

[To obtain a Source Code](#)

## About Q-Systems Engineering

TElasticForm component

### **Q-Systems Engineering**

Finite Elements Solutions, and Software and Engineering Consulting  
8950 N. Calle Buena Vista  
Tucson Az, 85704  
Tel: 520-742-7461; Fax:520-742-0402

**E-Mail: [qsystems@flash.net](mailto:qsystems@flash.net)**  
**<http://www.flash.net/~qsystems>**

# Installation

[TElasticForm](#) component

TElasticForm is provided as a zipped file ElastFrm.zip. Upon unzipping, it generates four local directories: Delphi1, Delphi 2, Delphi3, CBuilder, each of which contain the pertinent files for installation to the proper version of Delphi and C++Builder v.1. A fifth local directory Help is generated that includes the help file (Win 3.x help file).

Transfer the files that you need to your components directory and follow the simple component installation instructions of Delphi

[To Get Started](#)

## To Obtain a Registered Copy:

TElasticForm component

To obtain a registered copy of ElasticForm by e-mail :

**Credit Card Orders: (US\$15.00)**

Go to our general site : <http://www.flash.net/~qsystems>

or

ordering site: <http://order.kagi.com/?GDW>

and follow instructions.

**Pay by check: (US\$10.00)**

Send a certified check (personal checks from a U.S. Bank are also accepted but will take longer to clear) for \$10.00 to:

Claudia Peck  
8950 N. Calle Buena Vista  
Tucson, AZ 85704, U.S.A.

***Include your e-mail address***

***NOTE: The above price does NOT buy you the source code.***

Registration entitles you to free upgrades for one year from the time of purchase. It also rids your code of the initial message that appears on the screen every time your program uses a TElasticForm component.

**Please visit our site : <http://www.flash.net/~qsystems> to find out developing news on credit card purchasing.**

[To Get Started](#)

[To Obtain the Source Code](#)

## To Obtain the Source Code

TElasticForm component

### **Credit Card Orders: (US\$45.00)**

Go to our general site : <http://www.flash.net/~qsystems>

or

ordering site: <http://order.kagi.com/?GDW>

and follow instructions.

### **Pay by check: (US\$40.00)**

Send a certified check (personal checks from a U.S. Bank are also accepted but will take longer to clear) for \$40.00 to:

Claudia Peck  
8950 N. Calle Buena Vista  
Tucson, AZ 85704, U.S.A.

***Include your e-mail address***

**Please visit our site : <http://www.flash.net/~qsystems> to find out developing news on credit card purchasing.**

[To Get Started](#)

# TElasticForm component

[Properties](#)

[Methods](#)

[Tasks](#)

## Unit

ElasticForm

## Description

This component can do the following:

- Maintains the proportions of all visual components within a form when the form is resized.
- Maintains the size of a form upon change of resolution and/or change of font size.
- Sets allowed ranges for the size of the form
- Sets position of the form when maximized with set max limits smaller than those of the full screen.

Just drop the component of your form and design your form as if this component is not there.

[To Get Started](#)

## Properties

- ▶ Run-time only
- 🔑 Key properties

- 🔑 DesignClientFormHeight
- 🔑 DesignClientFormWidth
  - 🔑 DesignFormHeight
  - 🔑 DesignFormLeft
  - 🔑 DesignFormTop
  - 🔑 DesignFormWidth
  - 🔑 DesignPixelsPerInch
  - 🔑 DesignScreenHeight
  - 🔑 DesignScreenWidth
  - 🔑 ElasticFont
  - 🔑 ElasticHorizontal
  - 🔑 ElasticVertical
- 🔑 MaximizedPosX
  - 🔑 MaximizedPosY
  - 🔑 MaximumTrackHeight
  - 🔑 MaximumTrackWidth
  - 🔑 MinimumTrackHeight
  - 🔑 MinimumTrackWidth

## Methods

🔑 Key methods

Create

Destroy

UpdateSize

AddToExcludeList

# About the TElasticForm component

[TElasticForm reference](#)

## **Purpose**

TElasticForm is a nonvisual component to handle resizing of a form that contains visual components.

## **Tasks**

Just drop the component on the form.

## **Version**

Version 1.0

February 25, 1998

# DesignPixelsPerInch property

[See also](#)

## Applies to

[TElasticForm](#) component

## Declaration

```
property DesignPixelsPerInch: integer;
```

## Description

This is the value of Pixels Per Inch during design. Pixels per Inch is a value related to the size of the fonts. For example, small fonts corresponds to 96 Pixels per Inch, while large fonts corresponds to 120 Pixels per Inch.

This value is automatically read when you first drop the component on the form. If later you choose to change the font size during the desing stage, you must change the property manually on the Object Inspector or (recommended) delete the TElasticForm component from your form and drop on it a new one.

**See also**

[DesignScreenHeight](#), [DesignScreenWidth](#)

# DesignScreenHeight property

[See also](#)

## Applies to

[TElasticForm](#) component

## Declaration

```
property DesignScreenHeight: integer;
```

## Description

This is the value of Screen Height during design. Screen Height is the vertical screen resolution. Common values are 480, 600, 768, and 1024

This value is automatically read when you first drop the component on the form. If later you choose to change the screen resolution during the desing stage, you must change the property manually on the Object Inspector or (recommended) delete the TElasticForm component from your form and drop on it a new one.

**See also**

[DesignScreenWidth](#), [DesignPixelsPerInch](#)

# DesignScreenWidth property

[See also](#)

## Applies to

[TElasticForm](#) component

## Declaration

```
property DesignScreenWidth: integer;
```

## Description

This is the value of Screen Width during design. Screen Width is the horizontal screen resolution. Common values are 640, 800, 1024, and 1200.

This value is automatically read when you first drop the component on the form. If later you choose to change the screen resolution during the desing stage, you must change the property manually on the Object Inspector or (recommended) delete the TElasticForm component from your form and drop on it a new one.

**See also**

[DesignScreenHeight](#), [DesignPixelsPerInch](#)

## ElasticFont property

[See also](#)

### Applies to

[TElasticForm](#) component

### Declaration

```
property ElasticFont: boolean;
```

### Description

If ElasticFont is **true** then the fonts of all visual components of the form are resized as the form is resized. If it is **false** then the fonts maintain constant size during form resizing. The default value is **true**.

**See also**

[ElasticHorizontal](#), [ElasticVertical](#)

# ElasticHorizontal property

[See also](#)

## Applies to

[TElasticForm](#) component

## Declaration

```
property ElasticHorizontal: boolean;
```

## Description

If ElasticHorizontal is **true** then all visual components of the form are resized horizontally as the form is resized. If it is **false** then the components maintain constant horizontal size during form resizing. The default value is **true**.

**See also**

[ElasticFont](#), [ElasticVertical](#)

## ElasticVertical property

[See also](#)

### Applies to

[TElasticForm](#) component

### Declaration

```
property ElasticVertical: boolean;
```

### Description

If ElasticVertical is **true** then all visual components of the form are resized vertically as the form is resized. If it is **false** then the components maintain constant vertical size during form resizing. The default value is **true**.

**See also**

[ElasticFont](#), [ElasticHorizontal](#)

# MaximizedPosX property

[See also](#)

## Applies to

[TElasticForm](#) component

## Declaration

**property** MaximizedPosX: Integer;

## Description

MaximizedPosX is the horizontal coordinate of the top left corner of the form when the form is maximized. The default value is 0, which places the left side of the form flush with the left side of the screen. *If the value is provided as a positive integer then it is measured as pixels. If the value is provided as negative integer (between -1 and -100) then the absolute value of the number is interpreted as percentage of the scree size at run time.*

**See also**

[MaximizedPosY](#)

# MaximizedPosY property

[See also](#)

## Applies to

[TElasticForm](#) component

## Declaration

**property** MaximizedPosY: Integer;

## Description

MaximizedPosY is the vertical coordinate of the top left corner of the form when the form is maximized. The default value is 0, which places the top side of the form flush with the top of the screen. *If the value is provided as a positive integer then it is measured as pixels. If the value is provided as negative integer (between -1 and -100) then the absolute value of the number is interpreted as percentage of the screen size at run time*

**See also**

[MaximizedPosY](#)

# MaximumTrackHeight property

[See also](#)

## Applies to

[TElasticForm](#) component

## Declaration

**property** MaximumTrackHeight: Integer;

## Description

This is the maximum height of the form that is allowed either by resizing using the mouse or by maximizing pressing the maximize icon at the top right corner of the form. The default value is 0, which makes the maximum allowed height equal to the current vertical resolution of the screen. Setting MaximumTrackHeight equal to 500 prohibits the form from ever resizing to a height of 501 pixels or more. *If the value is provided as a positive integer then it is measured as pixels. If the value is provided as negative integer (between -1 and -100) then the absolute value of the number is interpreted as percentage of the scree size at run time*

**See also**

[MaximumTrackWidth](#), [MinimumTrackHeight](#), [MinimumTrackWidth](#)

# MaximumTrackWidth property

[See also](#)

## Applies to

[TElasticForm](#) component

## Declaration

**property** MaximumTrackWidth: Integer;

## Description

This is the maximum width of the form that is allowed either by resizing using the mouse or by maximizing pressing the maximize icon at the top right corner of the form. The default value is 0, which makes the maximum allowed width equal to the current horizontal resolution of the screen. Setting MaximumTrackWidth equal to 600 prohibits the form from ever resizing to a width of 601 pixels or more. *If the value is provided as a positive integer then it is measured as pixels. If the value is provided as negative integer (between -1 and -100) then the absolute value of the number is interpreted as percentage of the scree size at run time*

**See also**

[MaximumTrackHeight](#), [MinimumTrackHeight](#), [MinimumTrackWidth](#)

# MinimumTrackHeight property

[See also](#)

## Applies to

[TElasticForm](#) component

## Declaration

**property** MinimumTrackHeight: Integer;

## Description

This is the minimum allowed height of the form during resizing. The default value is 0, which allows the form to reduce to any size vertically. Setting MinimumTrackHeight equal to 200 prohibits the form from ever resizing to a height of 199 pixels or less. The intent of such an approach is so that the user cannot reduce your form to the extent that components are hidden or become too small to be recognizable or legible. *If the value is provided as a positive integer then it is measured as pixels. If the value is provided as negative integer (between -1 and -100) then the absolute value of the number is interpreted as percentage of the scree size at run time*

**See also**

[MaximumTrackHeight](#), [MaximumTrackWidth](#), [MinimumTrackWidth](#)

# MinimumTrackWidth property

[See also](#)

## Applies to

[TElasticForm](#) component

## Declaration

**property** MinimumTrackWidth: Integer;

## Description

This is the minimum allowed width of the form during resizing. The default value is 0, which allows the form to reduce to any size horizontally. Setting MinimumTrackWidth equal to 300 prohibits the form from ever resizing to a width of 299 pixels or less. The intent of such an approach is so that the user cannot reduce your form to the extent that components are hidden or become too small to be recognizable or legible. *If the value is provided as a positive integer then it is measured as pixels. If the value is provided as negative integer (between -1 and -100) then the absolute value of the number is interpreted as percentage of the scree size at run time*

**See also**

[MaximumTrackHeight](#), [MaximumTrackWidth](#), [MinimumTrackHeight](#)

# UpdateSize

## Applies to

[TElasticForm](#) component

## Declaration

```
procedure UpdateSize(aControl:TControl)
```

## Description

This is a procedure that should be used during run-time when a control is instructed to change size for reasons other than resizing. For example, consider a case that you chose to change the size of a button with name Button1 for emphasis. Then the change of size should be followed by the command UpdateSize(Button1). If this is not done then any resizing will give button1 its initial size.

# AddToExcludeList

## Applies to

[TElasticForm](#) component

## Declaration

```
procedure AddToExcludeList(aControl:TControl)
```

## Description

This is a procedure that should be used in the OnCreate procedure of the owner form to exclude the aControl visual component from resizing. This is a rarely needed procedure and should be used with caution. Non-resized components may overlap with resizable components during the resizing process.

## Example:

```
procedure TForm1.FormCreate(Sender: TObject);  
begin  
    TElasticForm1.AddToExcludeList(button1);  
end;
```

## DesignFormHeight property

### Applies to

[TElasticForm](#) component

This is the form height during design time. It is needed by the component. The user does not have to do anything about it.

# DesignFormLeft Property

## Applies to

[TElasticForm](#) component

This is the left coordinate of the form during design time. It is needed by the component. The user does not have to do anything about it.

What if the position of the form is saved in an INI file? [How do I override ElasticForm to make the form start at the saved position?](#)

# DesignFormTop Property

## Applies to

[TElasticForm](#) component

This is the top coordinate of the form during design time. It is needed by the component. The user does not have to do anything about it.

What if the position of the form is saved in an INI file? [How do I override ElasticForm to make the form start at the saved position?](#)

## DesignFormWidth Property

### Applies to

[TElasticForm](#) component

This is the form width during design time. It is needed by the component. The user does not have to do anything about it.

## DesignClientFormHeight

### Applies to

[TElasticForm](#) component

This is the form clientheight during design time. It is needed by the component. The user does not have to do anything about it.

## DesignClientFormWidth

### Applies to

[TElasticForm](#) component

This is the form clientwidth during design time. It is needed by the component. The user does not have to do anything about it.

## Bypassing the design form position

### Applies to

[TElasticForm](#) component

The idea here is that the user program may have saved the last position of the form in an INI file. Then next time the program should start at the last position rather than at the design position.

To achieve this, the **OnCreate** event of the form, should first read the saved left and top position of the form and then impose them to the Elasticform.

ElasticForm has three public (not published) properties that are designed to help in this direction:

```
ManualPosition : boolean;  
ManualLeft    : double;  
ManualTop     : double;
```

A problem may arise if the user saved the data under one resolution and then reloaded the program under a different resolution. In this case, the ElasticForm would redraw the form to maintain the same screen proportions, but would fail to place the form in the proper proportional position if left and top were saved in pixels values.

To bypass this problem, **ManualLeft** and **ManualTop** should be saved and read as percentages of the screen size at the time of save and read.

Here is a typical example that sets the form at: 5% of the width and 10% of the height of the screen:

```
procedure TForm1.FormCreate(Sender: TObject);  
begin  
  (*read ManualLeft and ManualTop from the INI file*)  
  with ElasticForm1 do  
    begin  
      ManualPosition := true;  
      ManualLeft := 0.05;  
      ManualTop := 0.10;  
    end;  
end;
```

