

# Table of Contents

Foreword	0
<b>Part I TFormHints - Overview</b>	<b>2</b>
<b>Part II Installation Instructions</b>	<b>2</b>
<b>Part III Registration Information</b>	<b>4</b>
<b>Part IV License Agreement</b>	<b>4</b>
<b>Part V Properties</b>	<b>6</b>
1 Active .....	6
2 BiDiMode .....	6
3 ColorBackground .....	7
4 ColorBorder .....	7
5 ColorShadow .....	7
6 Direction .....	8
7 Font .....	8
8 MaxWidth .....	8
9 Rounded .....	9
10 RoundRadius .....	9
11 ShowArrow .....	9
12 ShowShadow .....	10
13 TextAlignment .....	10
14 TransparencyRate .....	10
15 Transparent .....	11
<b>Part VI Events</b>	<b>11</b>
1 OnShowHint .....	11
<b>Index</b>	<b>0</b>

# 1 TFormHints - Overview

## Overview

The FormHints component applies the cartoon-style balloon hints for your controls instead of regular, rectangular hints. You may specify the default balloon [direction](#), [text alignment](#) and [degree of transparency](#) and other features that affect on hint appearance. One component changes hint appearance for all form's of entire application.

**new!** You can select the style of hint dynamically at run-time using [OnShowHint](#) event. For example you can select the cartoon-style hints for some certain controls but standard "boxy" style for all other controls.

**new!** Now you can use [MaxWidth](#) property to control the maximum width of balloon and wrap the long strings to next line.

## Basic properties

<a href="#">Active</a>	whether the FormHints is currently being in active state and can display balloon-like hints.
<a href="#">BiDiMode</a>	layout in right to left reading order for bi-directional text when the font selected is a Hebrew or Arabic font (bdRightToLeft).
<a href="#">ColorBackground</a> , <a href="#">ColorBorder</a> , <a href="#">ColorShadow</a>	specifies colors for various parts of balloon hint.
<a href="#">Direction</a>	specifies default direction for balloon.
<a href="#">MaxWidth</a>	specifies the maximum width of balloon. Allows to wrap the long strings to next line.
<a href="#">Rounded</a>	appearance of baloon corners. Can be rectangular or rounded.
<a href="#">ShowArrow</a> , <a href="#">ShowShadow</a>	whether the component should display arrow or shadow.
<a href="#">TextAlignment</a>	controls the text alignment within a balloon window
<a href="#">Transparent</a>	makes background of balloon transparent
<a href="#">TransparencyRate</a>	controls the degree of transparency for hint's background.

# 2 Installation Instructions

## Package without source code

### to Delphi 2

1. Unzip files from "Delphi2" directory to your "Delphi 2\Lib" directory.
2. Start Delphi 2 IDE.
3. Select "Component \ Install..." menu item.
4. Press "Add" button and select "FormHints.dcu" file.
5. Rebuild library.

### to Delphi 3

1. Unzip files from "Delphi3" directory and copy them to "Delphi 3\Lib".
2. Start Delphi 3 IDE.
3. Open "FormHintsD3.dpk" file.
4. Install package to the components palette ("Install" button).

### to Delphi 4

1. Unzip files from "Delphi4" directory and copy them to "Delphi 4\Lib".

2. Start Delphi 4 IDE.
3. Open "FormHintsD4.dpk" file.
4. Install package to the components palette ("Install" button).

#### to Delphi 5

1. Unzip files from "Delphi5" directory and copy them to "Delphi 5\Lib".
2. Start Delphi 5 IDE.
3. Open "FormHintsD5.dpk" file.
4. Install package to the components palette ("Install" button).

#### to Delphi 6

1. Unzip files from "Delphi6" directory and copy them to "Delphi 6\Lib".
2. Start Delphi 6 IDE.
3. Open "FormHintsD6.dpk" file.
4. Install package to the components palette ("Install" button).

#### to Delphi 7

1. Unzip files from "Delphi7" directory and copy them to "Delphi 7\Lib".
2. Start Delphi 7 IDE.
3. Open "FormHintsD7.dpk" file.
4. Install package to the components palette ("Install" button).

#### to C++ Builder 1

1. Unzip files from "BCB1" directory to your "CBuilder\Lib" directory.
2. Start C++ Builder IDE.
3. Select "Component \ Install..." menu item.
4. Press "Add" button and select "FormHints.dcu" file.
5. Rebuild library.

#### to C++ Builder 3

1. Unzip files from "BCB3" directory and copy them to "CBuilder3\Lib".
2. Start C++ Builder 3 IDE.
3. Open "FormHintsCB3.bpk" file.
6. Select "Project \ Make FormHintsCB3" menu item.
7. Select "Component \ InstallPackages" menu item.
8. Press "Add" button and select "FormHintsCB3.bpl" file.

#### to C++ Builder 4

1. Unzip files from "BCB4" directory and copy them to "CBuilder4\Lib".
2. Start C++ Builder 4 IDE.
3. Open "FormHintsCB4.bpk" file.
4. Install package to the components palette ("Install" button).

#### to C++ Builder 5

1. Unzip files from "BCB5" directory and copy them to "CBuilder5\Lib".
2. Start C++ Builder 5 IDE.
3. Open "FormHintsCB5.bpk" file.
4. Install package to the components palette ("Install" button).

#### to C++ Builder 6

1. Unzip files from "BCB6" directory and copy them to "CBuilder6\Lib".
2. Start C++ Builder 6 IDE.
3. Open "FormHintsCB6.bpk" file.
4. Install package to the components palette ("Install" button).

**Source code**

1. Uninstall / delete all previous (trial) instances of FormHints.
2. Unzip files from "Sources" directory and copy them to "..\Lib" directory.
3. Run Delphi or ++ Builder IDE.
4. Select "Component \ Install..." menu item.
5. Press "Add" button and select "FormHints.pas" file.
6. Rebuild library.

### 3 Registration Information

FormHints component is SHAREWARE. This means that you can try it out for free, but if you like it and want to use it you have to register it with the author. Before continue read and accept [license agreement](#) please.

The only difference between the unregistered and registered versions is that the registered one has not message box with remind to register and works without Delphi (C++ Builder) running. You can also purchase the [source code](#), if you would like to have it, and be able to compile or modify the FormHints on any 32bit version of Delphi or C++ Builder.

If you would like to use the FormHints and receive full, unrestricted version, priority support or even source code — you have to purchase proper license.

All prices in US dollars. Registering entitles you to unlimited support via E-Mail, minor version updates indefinitely and major version updates for 6 month from date of purchase.

**Registration types:*****Full, unrestricted version without source code:*****Single user license:**

- <https://secure.element5.com/register.html?productid=140830> - \$14,95

**Site license:**

- <https://secure.element5.com/register.html?productid=140831> - \$89,95

***Full version including 100% Source Code:*****Single user license:**

- <https://secure.element5.com/register.html?productid=140833> - \$24,95

**Site license:**

- <https://secure.element5.com/register.html?productid=140834> - \$139,95

**Comments**

1. **Site license** covers a single organisation in one location (building complex). If you buy a site license, you may use the software in unlimited number of your company's computers withing this area. Site license is very cost-effective if you have many computers (many software developers).

See [license agreement](#) for more details.

### 4 License Agreement

**Copyright**

The FormHints component (software) is Copyright © 1998-2002, by Utilmind Solutions® (Utilmind). All rights reserved.

The authors - Utilmind Solutions® and Aleksey Kuznetsov (founder of Utilmind), exclusively own all copyrights to the Advanced Application Controls (AppControls) and all other products distributed by Utilmind Solutions®.

**Liability disclaimer**

THIS SOFTWARE IS DISTRIBUTED "AS IS" AND WITHOUT WARRANTIES AS TO PERFORMANCE OF MERCHANTABILITY OR ANY OTHER WARRANTIES WHETHER EXPRESSED OR IMPLIED. YOU USE IT AT YOUR OWN RISK. THE AUTHOR WILL NOT BE LIABLE FOR DATA LOSS, DAMAGES, LOSS OF PROFITS OR ANY OTHER KIND OF LOSS WHILE USING OR MISUSING THIS SOFTWARE.

**Restrictions**

You may not attempt to reverse compile, modify, translate or disassemble the software in whole or in part. You may not remove or modify any copyright notice or the method by which it may be invoked.

**Operating license***Unregistered version*

You may distribute the unregistered version of software freely, provided that all files are included and remain unmodified and that no extra files have been added to the package. You may not ask any money for the distribution. You may use the unregistered version of software free of charge for testing purposes, but if you want to use it for other purposes than testing - you have to register it with the author.

*Registered version (single user license)*

Once you have registered, you will receive a personal registered copy via email and login information to access your personal area at AppControls.com. This copy may not be copied or lend. You have the non-exclusive right to use registered version of the software only by a single person, on a single computer at a time. You may physically transfer the software from one computer to another, provided that the software is used only by a single person, on a single computer at a time. In group projects where multiple persons will use the software, you must purchase an individual license for each member of the group or purchase site license. Use over a "local area network" (within the same locale) is permitted provided that the software is used only by a single person, on a single computer at a time. Use over a "wide area network" (outside the same locale) is strictly prohibited under any and all circumstances.

*Registered version (site/team license)*

Once you have registered, you will receive a personal registered copy via email and login information to access your personal area at AppControls.com. This copy may not be copied or lend. You have the non-exclusive right to use and transfer registered version of software on any number of computers by your company or your team only in one location (building complex). If you purchase a site license, you may use the program in an unlimited number of your company's computers within this area.

*Registered version (Educational site license)*

Once you have registered, you will receive a personal registered copy via email and login information to access your personal area at AppControls.com. This copy may not be copied or lend. You have the non-exclusive right to use and transfer registered version of software on any number of computers by your educational organisation (school/college/university etc) in one location (building complex). If you buy a educational site license, you may use the program in an unlimited number of your educational organisation's computers within this area.

*Registered version (World-wide license)*

Once you have registered, you will receive a personal registered copy via email and login information to access your personal area at AppControls.com. This copy may not be copied or lend. You have the non-exclusive right to use and transfer registered version of software on any number of computers by your company or your team world-wide. If your company has many branches even with thouthands of computers, world wide license covers them all.

Notes (clarification)

"Single-user license" means "single-developer license". "Site license" means that it can be used by any number of software developers within your company.

You can use purchased components in ANY number of your projects and deploy the "end-user" software to ANY number of your users/customers without any additional royalty fees. However you are not permitted to distribute the component itself (the source code or .dcu files of components).

**Back-up and transfer**

You may make one copy of the software solely for "back-up" purposes, as prescribed by international copyright laws. You must reproduce and include the copyright notice on the back-up copy.

**Terms**

This license is effective until terminated. You may terminate it by destroying the program, the documentation and copies thereof. This license will also terminate if you fail to comply with any terms or conditions of this agreement. You agree upon such termination to destroy all copies of the program and of the documentation, or return them to author.

**Other rights and restrictions**

All other rights and restrictions not specifically granted in this license are reserved by authors.

## 5 Properties

### 5.1 Active

**Applies to**

[FormHints](#) component.

**Declaration**

```
property Active: Boolean;
```

**Description**

The Active property controls whether the [FormHints](#) component is currently being in active state and can display balloon-like hints. If Active is True — FormHints component will hook all hint messages and show hints in balloon window. If Active is False — only regular hint boxes will be shown.

### 5.2 BiDiMode

**Applies to**

[FormHints](#) component.

**Declaration**

```
type
```

```
TfhSimpleBiDiMode = (bdLeftToRight, bdRightToLeft);
```

```
property BiDiMode: TfhSimpleBiDiMode;
```

**Description**

The BiDiMode property controls whether the text for balloon hints can be layout in right to left reading order for bi-directional text when the font selected is a Hebrew or Arabic font (bdRightToLeft). The default reading order is left to right (bdLeftToRight).

**See also**

[Font](#) property.

## 5.3 ColorBackground

**Applies to**

[FormHints](#) component.

**Declaration**

```
property ColorBackground: TColor;
```

**Description**

The ColorBackground property determines the background color for a balloon hints. Default value is clInfoBk.

**See also**

[ColorBorder](#), [ColorShadow](#) and [Font](#) properties.

## 5.4 ColorBorder

**Applies to**

[FormHints](#) component.

**Declaration**

```
property ColorBorder: TColor;
```

**Description**

The ColorBorder property determines the border color for a balloon hints. Default value is clBlack.

**See also**

[ColorBackground](#), [ColorShadow](#) and [Font](#) properties.

## 5.5 ColorShadow

**Applies to**

[FormHints](#) component.

**Declaration**

```
property ColorShadow: TColor;
```

**Description**

The ShadowColor property specifies the shadow color (dots that imitate the window shadow) for popup windows with context-sensitive help. ShadowColor used only if [ShowShadow](#) property is True.

**See also**

[ColorBackground](#), [ColorShadow](#) and [Font](#) properties.

## 5.6 Direction

### Applies to

[FormHints](#) component.

### Declaration

#### type

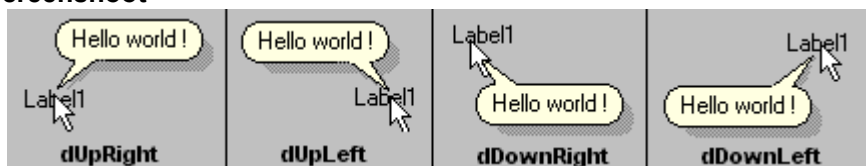
```
TfhHintDirection = (dUpRight, dUpLeft, dDownRight, dDownLeft);
```

**property** Direction: TfhHintDirection;

### Description

The Direction property determines default direction for balloon hint.

### Screenshoot



## 5.7 Font

### Applies to

[FormHints](#) component.

### Declaration

**property** Font: TFont;

### Description

The Font property is a font object that controls the attributes of hint text. To modify a font, you change the value of the Color, Name, Size, or Style properties of the font object.

### See also

[ColorBackground](#), [MaxWidth](#), [TextAlignment](#) and [BiDiMode](#) properties.

## 5.8 MaxWidth

### Applies to

[FormHints](#) component.


### Declaration

**property** MaxWidth: Word **default** 0;

### Description

The MaxWidth property used to control the maximum width of balloon-style hints and wrap long strings to next line, to fit the maximum width of balloon.

If MaxWidth is zero, the hints can have unlimited width, so lines are not wrapping.

 This property available since Delphi 3 / BCB 3 or higher. Unfortunately the wrapping of long hints is not allowed in Delphi 2.

### See also



[Font](#) and [TextAlignment](#) properties.

## 5.9 Rounded

### Applies to

[FormHints](#) component.

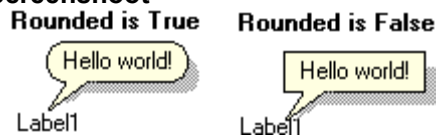
### Declaration

**property** Rounded: Boolean;

### Description

The Rounded property controls whether the corners of balloon hint should be rounded (if True), or rectangular as in regular hint window (if False).

### Screenshot



### See also

[RoundRadius](#) property.

## 5.10 RoundRadius

### Applies to

[FormHints](#) component.

### Declaration

**property** RoundRadius: Word;

### Description

The RoundRadius property determines the radius for rounded corners of balloon hints.

### See also

[Rounded](#) property.

## 5.11 ShowArrow

### Applies to

[FormHints](#) component.

### Declaration

**property** ShowArrow: Boolean;

### Description

The ShowArrow property controls whether the [FormHints](#) component should display the arrow for balloon hints.

### See also

[ShowShadow](#) property.

## 5.12 ShowShadow

### Applies to

[FormHints](#) component.

### Declaration

```
property ShowShadow: Boolean;
```

### Description

The ShowShadow property determines whether the [FormHints](#) should display a shadow (dots that imitate a shadow) behind a balloon window. You may also specify the color for shadow in [ColorShadow](#) property.

### See also

[ColorShadow](#) and [ShowArrow](#) properties.

## 5.13 TextAlignment

### Applies to

[FormHints](#) component.

### Declaration

```
type  
  TfhTextAlignment = (taNone, taLeft, taRight, taCenter);  
  
property TextAlignment: TfhTextAlignment;
```

### Description

The TextAlignment property controls how the text should be aligned within borders of balloon hint. When TextAlignment is taCenter, text will be centered within a balloon. When it taRight, text will be pressed to right edge of balloon.

### See also

[Font](#) and [MaxWidth](#) properties.

## 5.14 TransparencyRate

### Applies to

[FormHints](#) component.

### Declaration

```
type  
  TfhTransparencyRate = 0..100;  
  
property TransparencyRate: TfhTransparencyRate; // 0%..100%
```

### Description

The TransparencyRate property controls the degree of transparency for background of balloon hints. This property works if only Transparency is True. 100% transparent hints don't show the background at all.

### See also

[Transparent](#) property.

## 5.15 Transparent

### Applies to

[FormHints](#) component.

### Declaration

```
property Transparent: Boolean;
```

### Description

The Transparent property controls whether the background of balloon hints should be transparent. The [TransparencyRate](#) property controls the degree of transparency.

Transparency only take effect on balloon's [background](#). Text, border and shadow of balloon hint will displayed without transparency.

### See also

[TransparencyRate](#) and [ColorBackground](#) properties.

## 6 Events

### 6.1 OnShowHint

#### Applies to

[FormHints](#) component.

#### Declaration

##### type

```
TOnShowHint = procedure (Sender: TObject; HintControl: TControl; Hint: String; var AllowShowHint, ShowStandardHint: Boolean) of object;
```

```
property OnShowHint: TOnShowHint;
```

#### Description

The OnShowHint event occurs when application is about to show the hint. You can write this event handler if you would like to disallow hints for some controls at run-time (see *AllowShowHint* parameter). The control for which we should display the hint passed with *HintControl* parameter.

**new!** Now you can select the style of hint dynamically using *ShowStandardHint* parameter.

Following example demonstrates how to select cartoon-style hints only for TImages and standard style for other controls.

#### Example

```
procedure TForm1.acFormHints1ShowHint(Sender: TObject;
  HintControl: TControl; Hint: String; var AllowShowHint,
  ShowStandardHint: Boolean);
begin
  ShowStandardHint := not (HintControl is TImage);
end;
```