

Table of Contents

Foreword	0
Part I AnimationEffect	2
1 TAnimationEffect	2
2 Properties	2
AnimateOnShow	2
AnimateOnHide	2
Delay	3
EffectType	3
FireworkBlasts	3
Frames	3
Steps	4
3 Methods	4
Close	4
Open	4
Part II Win2kEffects	5
1 TWin2kEffects	5
2 Propeties	5
AnimateOnHide	5
Enabled	6
Flags	6
Time	7
AnimateOnShow	7
Enabled	8
Flags	8
Time	9
FlashCount	9
FlashTaskIcon	10
Translucent	10
TranslucentRate	10
3 Methods	11
Flash	11
FlashBackground	11
Part III Installation Instructions	12
Part IV Registration Information	13
Part V License Agreement	14
Index	0

1 AnimationEffect

1.1 TAnimationEffect

Overview

The AnimationEffect component will add a smooth animation effects on opening and closing of your windows and dialog boxes.

Just drop acAnimationEffect onto your form and make [AnimateOnShow](#) and [AnimateOnHide](#) properties True, and your form will appeared and disappeared with extremely cool animation effects without any line of code!

To choose definite animation effect - specify it in [EffectType](#) property. By default, component will show effects randomly.

To show animation effects manually - call [Open](#) and [Close](#) methods.

See also

[Win2kEffects](#) component.

1.2 Properties

1.2.1 AnimateOnShow

Applies to

[AnimationEffect](#) component.

Declaration

```
property AnimateOnShow: Boolean;
```

Description

The AnimateOnShow property controls whether you want to show [Open](#) Animation Effect (specified in [EffectType](#) property) during appearing window on screen.

See also

[AnimateOnHide](#) property.

1.2.2 AnimateOnHide

Applies to

[AnimationEffect](#) component.

Declaration

```
property AnimateOnHide: Boolean;
```

Description

The AnimateOnHide property controls whether you want to show [Close](#) Animation Effect (specified in [EffectType](#) property) during disappearing window from screen.

See also

[AnimateOnShow](#) property.

1.2.3 Delay

Applies to

[AnimationEffect](#) component.

Declaration

```
property Delay: Byte; // (CPU frequency * Delay) ticks
```

Description

The Delay property controls the delay in milliseconds between every [step](#) of animation. By default, Delay is 100 ticks.

1.2.4 EffectType

Applies to

[AnimationEffect](#) component.

Declaration**type**

```
TAnimationEffectType = (etRandom, etSpin, etVortex, etScatterGather,  
etSpike, etFireworks);
```

```
property EffectType: TAnimationEffectType;
```

Description

The EffectType property specifies the type of AnimationEffect, which will be played on calling the [Open](#) or [Close](#) methods.

Besides the type of effect you may specify the number of animation [Frames](#) and [Steps](#).

If EffectType is etRandom, AnimationEffect component will show random effect - Spin, Vortex, ScatterGather, Spike or Fireworks. You may also specify number of [FireworkBlasts](#).

1.2.5 FireworkBlasts

Applies to

[AnimationEffect](#) component.

Declaration

```
property FireworkBlasts: Byte;
```

Description

The FireworkBlasts property specifies the number of blast-points for Firework animation effect. Default value is 10 blasts. See [EffectType](#) property for more info.

1.2.6 Frames

Applies to

[AnimationEffect](#) component.

Declaration

```
property Frames: Byte;
```

Description

The Frames property specifies the number of animation frames. Default value is 6.

See also

[EffectType](#) and [Steps](#) properties.

1.2.7 Steps

Applies to

[AnimationEffect](#) component.

Declaration

```
property Steps: Byte;
```

Description

The Steps property specifies the number of steps on drawing the animation effect. Default value is 50 steps.

See also

[EffectType](#) and [Frames](#) properties.

1.3 Methods

1.3.1 Close

Applies to

[AnimationEffect](#) component.

Declaration

```
procedure Close;
```

Description

The Close method draws smooth animation effect which simulate closing of window.

See also

[EffectType](#) property and example of usage of the Close method.

1.3.2 Open

Applies to

[AnimationEffect](#) component.

Declaration

```
procedure Open;
```

Description

The Open method draws smooth animation effect which simulate opening of window.

See also

[EffectType](#) property and example of usage of the Open method.

2 Win2kEffects

2.1 TWin2kEffects

Overview

The Win2kEffects component enables you to produce some special effects of the Windows 98 / 2000. You can show roll, slide and alpha-blending animation effects on showing and hiding windows, make your forms translucent, flash the caption bars and taskbar icons.

Rolling / sliding animation and flashing effect works on Windows 98 and later. Blending animation and translucent forms supported in Windows 2000 and later. You can specify any of these effects on Windows 95 / Windows NT4 systems, but no effects will be shown.

Animation effects ?

You can produce special effects when showing ([AnimateOnShow](#) structure) or hiding windows ([AnimateOnHide](#) structure). There is three types of animation: roll, slide and alpha-blending. See possible animation [Flags](#) for more details.

Translucent forms ?

The partially translucent, alpha-blended forms supported in Windows 2000. If you would like to make the form translucent — set [Translucent](#) property to True and specify the rate of alpha-blending in the [TranslucentRate](#) property.

Flashing windows ?

The Win2kEffects component can flash the window caption and the application icon on task bar. Typically, you flash a window to inform the user that the window requires attention but does not currently have the keyboard focus. When a window flashes, it appears to change from inactive to active status. An inactive caption bar changes to an active caption bar; an active caption bar changes to an inactive caption bar. To flash the form — use [Flash](#) method. Number of times to flash the window specified in the [FlashCount](#) property.

See also

[AnimationEffect](#) component.

2.2 Properties

2.2.1 AnimateOnHide

Applies to

[Win2kEffects](#) and [AnimationEffect](#) components.

Declaration

type

```
TWin2kAnimation = class(TPersistent)
published
  property Enabled: Boolean;
  property Flags: TWin2kAnimationFlags; // kind of animation
  property Time: Word; // duration of animation
end;
```

```
property AnimateOnHide: TWin2kAnimation;
```

Description

The AnimateOnHide structure controls the behaviors of the Win2k-based animation effect which will be shown when the form is about to disappear from the screen.

See also

[AnimateOnShow](#) property.

2.2.1.1 Enabled**Applies to**

[Win2kEffects](#) as subproperty of [AnimateOnShow](#) and [AnimateOnHide](#) structures.

Declaration

```
property Enabled: Boolean;
```

Description

The Enabled property controls whether you would like to show the animation effect on hiding the form. Kind of the animation effect can be specified in the [Flags](#) property.

See also

[AnimateOnShow](#) and [AnimateOnHide](#) structures.

2.2.1.2 Flags**Applies to**

[Win2kEffects](#) as subproperty of [AnimateOnShow](#) and [AnimateOnHide](#) structures.

Declaration**type**

```
TWin2kAnimationFlags = set of (afHorPositive, afHorNegative,
afVerPositive, afVerNegative, afCenter, afSlide, afBlend);
```

```
property Flags: TWin2kAnimationFlags;
```

Description

Specifies the type of animation. This parameter can be one of the following values:

Value	Description
afHorPositive	Animates the window from left to right. This flag can be used with roll or slide animation. It is ignored when used with afCenter or afBlend flags.
afHorNegative	Animates the window from right to left. This flag can be used with roll or slide animation. It is ignored when used with afCenter or afBlend flags.
afVerPositive	Animates the window from top to bottom. This flag can be used with roll or slide animation. It is ignored when used with afCenter or afBlend flags.
afVerNegative	Animates the window from bottom to top. This flag can be used with roll or slide animation. It is ignored when used with afCenter or afBlend flags.
afCenter	Makes the window appear to collapse inward when form is hiding or expand outward if form is showing.
afSlide	Uses slide animation. By default, roll animation is used. This flag is ignored when used with afCenter.
afBlend	Uses a fade effect. This flag works in Win2000 or later (<i>does not work in Win95/98/NT4!</i>) and can be used only if you animating a top-level window.

Short tutorial

To show the rolling effect from left to right set the Flags property to [afHorPositive]:

```
Win2kEffects1.AnimateOnHide.Flags := [afHorPositive];
```

To show sliding effect instead of rolling, just add the afSlide flag. For example, to show sliding effect from upper-rightcorner to bottom-left corner, set the Flags property to [afHorNegative,afVerPositive,afSlide]:

```
Win2kEffects1.AnimateOnHide.Flags := [afHorNegative, afVerPositive,
    afSlide];
```

To show the alpha-blendingeffect, set the Flags property to [afBlend]:

```
Win2kEffects1.AnimateOnHide.Flags := [afBlend];
```

🚩 Alpha-blendingworks on Windows 2000 and later only. If you will try to use blending effect on Windows 98, nothing will happends. When you're using fade effect, you don't need to specify any others flags.

See also

[AnimateOnShow](#) and [AnimateOnHide](#) structures.

2.2.1.3 Time

Applies to

[Win2kEffects](#) as subproperty of [AnimateOnShow](#) and [AnimateOnHide](#) structures.

Declaration

```
property Time: Word; // Duration of animation. Default 200
```

Description

The Time property specifies how long it takes to play the animation, in milliseconds. Typically, an animation takes 200 milliseconds to play.

See also

[AnimateOnShow](#) and [AnimateOnHide](#) structures.

2.2.2 AnimateOnShow

Applies to

[Win2kEffects](#) and [AnimationEffect](#) components.

Declaration

type

```
TWin2kAnimation = class(TPersistent)
```

published

```
property Enabled: Boolean;
```

```
property Flags: TWin2kAnimationFlags; // kind of animation
```

```
property Time: Word; // duration of animation
```

```
end;
```

```
property AnimateOnShow: TWin2kAnimation;
```

Description

The AnimateOnShow structure controls the behaviorof the Win2k-based animation effect which will be shown when the form is about to appear on the screen.

See also

[AnimateOnHide](#) property.

2.2.2.1 Enabled**Applies to**

[Win2kEffects](#) as subproperty of [AnimateOnShow](#) and [AnimateOnHide](#) structures.

Declaration

```
property Enabled: Boolean;
```

Description

The Enabled property controls whether you would like to show the animation effect on showing the form. Kind of the animation effect can be specified in the [Flags](#) property.

See also

[AnimateOnShow](#) and [AnimateOnHide](#) structures.

2.2.2.2 Flags**Applies to**

[Win2kEffects](#) as subproperty of [AnimateOnShow](#) and [AnimateOnHide](#) structures.

Declaration**type**

```
TWin2kAnimationFlags = set of (afHorPositive, afHorNegative,
afVerPositive, afVerNegative, afCenter, afSlide, afBlend);
```

```
property Flags: TWin2kAnimationFlags;
```

Description

Specifies the type of animation. This parameter can be one of the following values:

<u>Value</u>	<u>Description</u>
afHorPositive	Animates the window from left to right. This flag can be used with roll or slide animation. It is ignored when used with afCenter or afBlend flags.
afHorNegative	Animates the window from right to left. This flag can be used with roll or slide animation. It is ignored when used with afCenter or afBlend flags.
afVerPositive	Animates the window from top to bottom. This flag can be used with roll or slide animation. It is ignored when used with afCenter or afBlend flags.
afVerNegative	Animates the window from bottom to top. This flag can be used with roll or slide animation. It is ignored when used with afCenter or afBlend flags.
afCenter	Makes the window appear to collapse inward when form is hiding or expand outward if form is showing.
afSlide	Uses slide animation. By default, roll animation is used. This flag is ignored when used with afCenter.
afBlend	Uses a fade effect. This flag works in Win2000 or later (<i>does not work in Win95/98/NT4!</i>) and can be used only if you animating a top-level window.

Short tutorial

To show the rolling effect from left to right set the Flags property to [afHorPositive]:

```
Win2kEffects1.AnimateOnShow.Flags := [afHorPositive];
```


To show sliding effect instead of rolling, just add the `afSlide` flag. For example, to show sliding effect from upper-right corner to bottom-left corner, set the `Flags` property to `[afHorNegative, afVerPositive, afSlide]`:

```
Win2kEffects1.AnimateOnShow.Flags := [afHorNegative, afVerPositive, afSlide];
```

To show the alpha-blending effect, set the `Flags` property to `[afBlend]`:

```
Win2kEffects1.AnimateOnShow.Flags := [afBlend];
```

💡 Alpha-blending works on Windows 2000 and later only. If you will try to use blending effect on Windows 98, nothing will happen. When you're using fade effect, you don't need to specify any other flags.

See also

[AnimateOnShow](#) and [AnimateOnHide](#) structures.

2.2.2.3 Time

Applies to

[Win2kEffects](#) as subproperty of [AnimateOnShow](#) and [AnimateOnHide](#) structures.

Declaration

```
property Time: Word; // Duration of animation. Default 200
```

Description

The `Time` property specifies how long it takes to play the animation, in milliseconds. Typically, an animation takes 200 milliseconds to play.

See also

[AnimateOnShow](#) and [AnimateOnHide](#) structures.

2.2.3 FlashCount

Applies to

[Win2kEffects](#) component.

Declaration

```
property FlashCount: Word; // Default 4
```

Description

The `FlashCount` property specifies the number of times to flash the window. To flash the window and/or [task icon](#) — use [Flash](#) method.

Requirements

Windows NT/2000: Requires Windows 2000 or later.

Windows 95/98: Requires Windows 98 or later.

💡 If you call this function on Win95/NT4 — nothing will happen.

See also

[FlashTaskIcon](#) property and [Flash](#) method.

2.2.4 FlashTaskIcon

Applies to

[Win2kEffects](#) component.

Declaration

```
property FlashTaskIcon: Boolean;
```

Description

The FlashTaskIcon property controls whether you would like to flash the application icon on the taskbar at the same time with the caption. Set FlashTaskIcon to True to flash the task icon or False to flash only window caption.

The number of times to flash the window can be specified in the [FlashCount](#) property.

Requirements

Windows NT/2000: Requires Windows 2000 or later.

Windows 95/98: Requires Windows 98 or later.



If you call this function on Win95/NT4— nothing will happen.

See also

[FlashCount](#) property and [Flash](#) method.

2.2.5 Translucent

Applies to

[Win2kEffects](#) component.

Declaration

```
property Translucent: Boolean;
```

Description

The Translucent property controls whether the form is translucent, that is, alpha-blended. Set Translucent to True to make the form partially translucent, by rate specified in the [TranslucentRate](#) property.



The translucent forms supported by Windows 2000 and later. You can specify this property in lower Windows versions but nothing will happen.

See also

[TranslucentRate](#) property.

2.2.6 TranslucentRate

Applies to

[Win2kEffects](#) component.


Declaration**type**

```
TWin2kTranslucentRate = 0..100;
```

```
property TranslucentRate: TWin2kTranslucentRate; // 0%..100%
```

Description

The `TransluentRate` controls the transparency rate, in percents, of the alpha-blended form, when the `Transluent` property is `True`.

 The translucent forms supported by Windows 2000 and later. You can specify this property in lower Windows versions but nothing will happen.

See also

`Transluent` property.

2.3 Methods

2.3.1 Flash

Applies to

`Win2kEffects` component.


Declaration

```
procedure Flash;
```

Description

The `Flash` method flashes the caption of current form + the taskbar icon if the `FlashTaskIcon` property is `True`. Number of times to flash the window can be specified in the `FlashCount` property.


The `Flash` method flashes the window but does not change the active state of the window.

 Typically, you flash a window to inform the user that the window requires attention but does not currently have the keyboard focus. When a window flashes, it appears to change from inactive to active status. An inactive caption bar changes to an active caption bar; an active caption bar changes to an inactive caption bar.

Requirements

Windows NT/2000: Requires Windows 2000 or later.

Windows 95/98: Requires Windows 98 or later.

 If you call this function on Win95/NT4 — nothing will happen.

See also

`FlashBackground` method;
`FlashCount`, `FlashTaskIcon` properties.

2.3.2 FlashBackground

Applies to


`Win2kEffects` component.

Declaration

```
procedure FlashBackground;
```

Description

The `FlashBackground` method flashes the caption and window icon of current form *continuously*, until the window comes to the foreground.

 Typically, you flash a window to inform the user that the window requires attention but does not currently have the keyboard focus. When a window flashes, it appears to change from inactive to active status. An inactive caption bar changes to an active caption bar; an active caption bar changes to an inactive caption bar.

Requirements

Windows NT/2000: Requires Windows 2000 or later.

Windows 95/98: Requires Windows 98 or later.

See also

[Flash](#) method.

3 Installation Instructions

Package without source code

to Delphi 2

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

to Delphi 3

1. Unzip files from "Delphi3" directory and copy them to "Delphi 3\Lib".
2. Start Delphi3 IDE.
3. Open "AnimationEffectD3.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 Delphi4 IDE.
3. Open "AnimationEffectD4.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 Delphi5 IDE.
3. Open "AnimationEffectD5.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 "AnimationEffectD6.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 "AnimationEffectD7.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 "AnimationEffect.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 "AnimationEffectCB3.bpk" file.
6. Select "Project \ Make AnimationEffectCB3" menu item.
7. Select "Component\ InstallPackages" menu item.
8. Press "Add" button and select "AnimationEffectCB3.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 "AnimationEffectCB4.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 "AnimationEffectCB5.bpk" file.
4. Install package to the components palette ("Install" button).

Source code

1. Uninstall / delete all previous(trial) instances of AnimationEffect.
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 "_AEReg.pas" file.
6. Rebuild library.

4 Registration Information

AnimationEffect 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 AnimationEffect on any 32bit version of Delphi or C++ Builder.

If you would like to use the AnimationEffect 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 developer license:**

- <https://secure.element5.com/register.html?productid=140725> - \$14,95

Site license:

- <https://secure.element5.com/register.html?productid=140726> - \$89,95

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

- <https://secure.element5.com/register.html?productid=140727> - \$27,95

Site license:

- <https://secure.element5.com/register.html?productid=140728> - \$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 within this area. Site license is very cost-effective if you have many computers (many software developers).

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

5 License Agreement

Copyright

The AnimationEffect 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 thousands 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.