

---

# FreeEsVclComponents & EsVclCore

---

Бесплатная библиотека VCL компонентов для  
Delphi/C++Builder

---



[github.com/errorcalc](https://github.com/errorcalc)  
[errorsoft@mail.ru](mailto:errorsoft@mail.ru)

## Оглавление

Структура .....	2
TEsCustomControl .....	3
TEsSwitch.....	5

## Структура

**FreeEsVclComponents** – бесплатная библиотека компонентов для Delphi/C++Builder.

Состоит из нескольких пакетов:

- **EsVclCore** – основные классы/функции/типы
- **FreeEsVclComponents** – непосредственно компоненты
- **FreeEsVclComponentsDesign** – вспомогательный Design-time пакет

## TEsCustomControl

```
TEsCustomControl = class(TWinControl)
private
    // anti flicker and transparent magic
    FCanvas: TCanvas;
    CacheBitmap: HBITMAP; // Cache for buffer BitMap
    CacheBackground: HBITMAP; // Cache for background BitMap
    FIsCachedBuffer: Boolean;
    FIsCachedBackground: Boolean;
    StoredCachedBuffer: Boolean;
    StoredCachedBackground: Boolean;
    FBufferedChildrens: Boolean;
    FParentBufferedChildrens: Boolean;
    FIsFullSizeBuffer: Boolean;
    // paint events
    FOnPaint: TPaintEvent;
    FOnPainting: TPaintEvent;
    // draw helper
    FIsDrawHelper: Boolean;
```

**Описание**

Базовый класс, содержит в себе «магию» подавления мерцания и прозрачности.  
Это аналог TCustomControl, но с расширенными возможностями.

**Свойства**

property Canvas: TCanvas read FCanvas;

Аналогично TCustomControl

property BufferedChildrens: Boolean read FBufferedChildrens write  
SetBufferedChildrens stored IsBufferedChildrensStored;

Буферизация внутренних компонентов, которые унаследованы от TGraphicControl.  
Очень полезное свойство, позволяющее устранить мерцание у внутренних компонентов, таких  
как TLabel, TImage, TShape...

property ParentBufferedChildrens: Boolean read FParentBufferedChildrens  
write SetParentBufferedChildrens default True;

Значение BufferedChildrens будет копировать родительское.

property IsCachedBuffer: Boolean read FIsCachedBuffer write  
SetIsCachedBuffer default False;

Позволяет указать, что BitMap для двойной буферизации не должен удаляться после  
перерисовки, это ускоряет прорисовку, однако количество потребляемой памяти увеличивается.

property IsCachedBackground: Boolean read FIsCachedBackground write  
SetIsCachedBackground default False;

Кэширование фона, увеличивает скорость прорисовки, однако количество потребляемой памяти  
увеличивается.

См. procedure UpdateBackground

property IsDrawHelper: Boolean read FIsDrawHelper write SetIsDrawHelper  
default False;

Прорисовка «хелпера» в design-time, по умолчанию это просто прорисовка границы контроля,  
однако наследники могут рисовать дополнительные границы.

```
property IsOpaque: Boolean read GetIsOpaque write SetIsOpaque default  
False;
```

Непрозрачность контрола, при установке в True контрол становится абсолютно непрозрачен.  
Кроме того контрол перестаёт прорисовывать фон.  
Необходимо использовать если вы полностью прорисовываете контрол.

```
property IsFullSizeBuffer: Boolean read FIsFullSizeBuffer write  
FIsFullSizeBuffer default False;
```

Bitmap двойной буферизации всегда создаётся по размеру всего контрола, даже при частичной перерисовке.

## События

```
property OnPainting: TPaintEvent
```

Событие вызывается перед прорисовкой компонента.

```
property OnPaint: TPaintEvent
```

Событие вызывается после прорисовки компонента.

## Функции

```
procedure UpdateBackground(Repaint: Boolean); overload;
```

```
procedure UpdateBackground; overload;
```

Обновляет кэш с фоном, имеет смысл при IsCachedBackground = False

## Расположение

ES.BaseControls.pas

## TEsSwitch



### Описание

Компонент-переключатель, имитирует одноименный XAML контрол Windows 10.  
Имеет поддержку ActionList.

### Свойства

```
property Alignment: TSwitchAlignment read FAlignment write SetAlignment  
default saRight;
```

Расположение переключателя.

```
property FrameColor: TColor read GetFrameColor write SetFrameColor  
default clDefault;
```

Цвет границы переключателя. По умолчанию установлен в clDefault, что означает что  
используются системный цвет.

```
property ThumbColor: TColor read GetThumbColor write SetThumbColor  
default clDefault;
```

Цвет ползунка переключателя. По умолчанию установлен в clDefault, что означает что  
используются системный цвет.

```
property MainColor: TColor read GetMainColor write SetMainColor default  
clDefault;
```

Главный цвет переключателя. По умолчанию установлен в clDefault, что означает что  
используются системный цвет. (Для Windows Vista и далее используется DwmGetColorizationColor)

```
property Checked: Boolean read FChecked write SetChecked default False;
```

Состояние компонента включен\выключен.

```
property TextOn: string read FTextOn write SetTextOn stored  
IsStoredTextOn;
```

```
property TextOff: string read FTextOff write SetTextOff stored  
IsStoredTextOff;
```

Текст включённого\выключенного контроля.

```
property ShowCaption: Boolean read FShowCaption write SetShowCaption  
default False;
```

Включение\выключение отображения текста.

```
property Animated: Boolean read FAnimated write FAnimated default True;
```

Включение\выключение анимации.

```
property SwitchWidth: Integer read FSwitchWidth write SetSwitchWidth  
default DefaultWidth;
```

```
property SwitchHeight: Integer read FSwitchHeight write SetSwitchHeight  
default DefaultHeight;
```

Ширина\высота переключателя.

```
property SwitchLayout: TSwitchLayout read FSwitchLayout write  
    SetSwitchLayout default slFixed;
```

Расположение и размер переключателя.

```
property AutoSize: Boolean read FAutoSize write SetAutoSize default True;
```

Автоматическое изменение размера.

## **Расположение**

ES.Switch.pas