



Компонент **TComboBox** сочетает в себе редактируемый список с возможностью прокрутки и выбора определенного пункта списка. Пользователь может выбрать какой-либо элемент из списка или ввести его непосредственно в поле редактирования.

### Раскрывающийся список

Когда список находится в открытом состоянии, Windows рисует окно внизу списка с элементами, которые находятся внутри списка.

Свойство *DropDownCount* определяет максимальное количество элементов, которые будут отображаться в открытом списке.

**Ширина открытого списка равна ширине поля ComboBox.** Когда длина (или строка) превышает ширину **ComboBox**, элементы урезаются до ширины списка.

Компонент TComboBox не имеет свойства для расширения раскрывающегося окна списка.

### Фиксированная ширина раскрывающегося поля ComboBox

Но можно самостоятельно установить ширину раскрывающегося списка, отправив специальное сообщение Windows **ComboBox**. Это сообщение **CB\_SETDROPPEDWIDTH**, которое означает минимально допустимую ширину раскрывающегося списка.

Для размера окна, скажем, 200 пикселей, можно сделать так:

```
SendMessage(theComboBox.Handle, CB_SETDROPPEDWIDTH, 200, 0);
```

Это будет работать правильно, если только ширина пунктов раскрывающегося списка не

будет превышать 200 пикселей.

Для того, чтобы быть уверенным, что пункты меню "влезут" в раскрывающийся список, можно будет сделать вычисления.

Вот пример функции для расчета ширины раскрывающегося списка с учетом ширины его отдельных пунктов.

```
procedure ComboBox_AutoWidth(const theComboBox: TCombobox); const
HORIZONTAL_PADDING = 4; var itemsFullWidth: integer; idx: integer; itemWidth:
integer; begin itemsFullWidth := 0; // получить ширину пунктов в открытом состоянии
for idx := 0 to -1 + theComboBox.Items.Count do begin itemWidth :=
theComboBox.Canvas.TextWidth( theComboBox.Items[idx]); Inc(itemWidth, 2 *
HORIZONTAL_PADDING); if (itemWidth > itemsFullWidth) then itemsFullWidth :=
itemWidth; end; // установить ширину, если необходимо if (itemsFullWidth >
theComboBox.Width) then begin // проверить ширину, если есть полоса прокрутки if
theComboBox.DropDownCount > 0 then begin // для расчета ширины раскрывающегося списка
используется ширина самой длинной строки.
```

### Откуда вызывать процедуру `ComboBox_AutoWidth`?

Если Вы предварительно заполняете список во время разработки, то вызывать процедуру `ComboBox_AutoWidth` удобно при создании формы, в обработчике события `OnCreate` формы.

Если же Вы динамически заполняете список во время работы приложения, то вызывать процедуру `ComboBox_AutoWidth` можно внутри обработчика события `OnDropDown` `ComboBox`. Это событие происходит, когда Вы открываете раскрывающийся список.

### Проверяем

Для проверки, поместите компонент `TComboBox` на форму и заполните его свойство `Items` словами различной длины. Затем пишем процедуру `ComboBox_AutoWidth`

и вызываем ее в событии формы

### OnCreate

```
procedure TForm1.FormCreate(Sender: TObject); begin  
  ComboBox1.AutoSize(ComboBox1); end;
```

После запуска программы, Вы можете убедиться, что ширина раскрывающегося списка растянется по ширине самого длинного пункта списка.

### Расширение списка влево!

Заметьте, что список расширяется вправо. Что же сделать, если **ComboBox** находится справа на форме, а поле со списком выходит за пределы окна?

Сообщение **CB\_SETDROPPEDWIDTH** всегда расширяет список вправо и нет никакой возможности указать, в каком направлении нужно расширить список (вправо или влево).

### Найденное решение - WM\_CTLCLORLISTBOX

Решение простое: когда открывается список, Windows отправляет сообщение **WM\_CTLCLORLISTBOX**

Каждое **VCL** управление предоставляет свойства **WindowProc** - это процедура, которая реагирует на сообщения, которые отправляются на управление. Мы можем использовать свойство

### WindowProc

, чтобы временно заменить, или сделать подкласс процедуры окна управления.

Вот наш измененный **WindowProc** для **ComboBox**:

```
// измененный WindowProc  
procedure TForm1.ComboBox1WindowProc( var Message: TMessage);  
var cr, lbr: TRect; begin // прорисовываем окно списка  
  if Message.Msg = WM_CTLCLORLISTBOX then begin  
    GetWindowRect(ComboBox1.Handle, cr); //  
    GetWindowRect(Message.LParam, lbr); // перемещаем влево,  
    // выравниваем по правому краю  
    if cr.Right > lbr.Right then
```

```
MoveWindow(Message.LParam,          lbr.Left-(lbr.Right-cr.Right),          lbr.Top,
           lbr.Right-lbr.Left,      lbr.Bottom-lbr.Top,          True); end else
ComboBox1WindowProcORIGINAL(Message); end;
```

Когда наш список получит сообщение **WM\_CTLCOLORLISTBOX**, мы получим размеры окна списка, а также прямоугольник окна раскрытого списка (

### **GetWindowRect**

). Если окажется, что раскрывающийся список будет правее самого

### **ComboBox**

, то мы переместим его левее, т.е. выровняем по правому краю.

Если же сообщение не **WM\_CTLCOLORLISTBOX**, то мы просто вызываем оригинальную процедуру обработки сообщения.

Ну и наконец, нужно разместить все это в обработчике события **OnCreate** формы:

```
procedure TForm1.FormCreate(Sender: TObject); begin
ComboBox_AutoWidth(ComboBox1); // прикрепляем
модифицированный/пользовательский // WindowProc для ComboBox3
ComboBox1WindowProcORIGINAL := ComboBox1.WindowProc; ComboBox1.WindowProc
:= ComboBox1WindowProc; end;
```

И конечно же прописать все в объявлении формы:

```
type TForm1 = class(TForm)  ComboBox1: TComboBox;  Button1: TButton;
procedure FormCreate(Sender: TObject);  procedure Button1Click(Sender: TObject);
private { Private declarations }  ComboBox1WindowProcORIGINAL: TWndMethod;
procedure ComboBox1WindowProc( var Message: TMessage);  public { Public
declarations } end;
```

Вот и все! И все это работает!