

**ComboBox** Windows содержит список. В стандартном **ComboBox** этот список имеет такую же ширину, что и сам

### **ComboBox**

. Можно изменить ширину в большую или меньшую сторону. И даже можно изменить позицию, где этот список будет появляться. Вы, возможно, видели, как это сделано в Microsoft Word и Microsoft Excel.

**ComboBox** в Windows - это фактически комбинация из двух или более управлений, именно поэтому он и называется *Combo*.

Чтобы изменить вид списка, нужен дескриптор управления списка. Это трудно, потому что список это дочернее окно рабочего стола (для стилей **CBS\_DROPDOWN** и **CBS\_DROPDOWNLIST**).

**ComboBox** получает сообщения **WM\_CTLCLORMSGBOX** .. **WM\_CTLCLORSTATIC** для его составляющего управления, когда они должны быть прорисованы. Это позволяет

### **Combobox**

определять цвет для этого управления. В случае

### **ComboBox**

, Windows посылает сообщение

### **WM\_CTLCLORLISTBOX**

с параметром

*lParam*

, который содержит дескриптор управления списка.

Как только мы получим дескриптор управления списка, мы можем изменить размеры управления, используя **MoveWindow** API.

Код ниже демонстрирует, как сделать это в новом компоненте:

```
unit siComboBox; interface uses Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs, StdCtrls; type TsiComboBox = class(TComboBox) private {
```

```
Private declarations } FFirst: boolean; FRect: TRect; FListHandle: HWND;
FCWX: integer; FCX: integer; FCY: integer; protected { Protected declarations } //
мы должны отменить это для измененного поведения procedure WndProc(var Message:
TMessage); override; public { Public declarations } constructor Create(AOwner:
TComponent); override; published { Published declarations } // Свойства ширины,
высоты и левой позиции раскрывающегося списка: property CWX: integer read FCWX
write FCWX default 0; property CX: integer read FCX write FCX default 0; property CY:
integer read FCY write FCY default 0; end; procedure Register; implementation
procedure Register; begin RegisterComponents('siComponents', [TsiComboBox]); end; {
TsiComboBox } constructor TsiComboBox.Create(AOwner: TComponent); begin inherited;
FFirst := True; FCX := 0; FCWX := 0; FCY := 0; end; procedure
TsiComboBox.WndProc(var Message: TMessage); Var CurRect: TRect; begin inherited;
if Message.Msg = WM_CTLCOLORLISTBOX then begin // Получаем дескриптор списка
FListHandle := Message.LParam; // Получаем прямоугольник списка GetWindowRect
(FListHandle, CurRect); If (CurRect.Left FRect.Left) Or (CurRect.Top FRect.Top) Then
Begin FRect := CurRect; End; // Изменяем размеры списка MoveWindow (
FListHandle, FRect.left + FCX, FRect.top + FCY, (FRect.right - FRect.left +
FCWX), FRect.bottom - FRect.top, TRUE ); end; end; end.
```

Если Вы только хотите расширить раскрывающийся список **ComboBox**, Вы можете использовать сообщение **CB\_SETDROPPEDWIDTH** в событии **OnDropDown**. Например:

```
procedure TForm1.ComboBox1DropDown(Sender: TObject); var MyListWidth : Integer;
begin MyListWidth := 100; // Вы можете вычислить значение.
SendMessage(Combobox1.Handle, CB_SETDROPPEDWIDTH, MyListWidth, 0); end;
```

Если **MyListWidth** меньше ширины **ComboBox**, это игнорируется.