

Эта статья описывает, как добавить **TextCompletion** в **ComboBox**, создавая новый визуальный компонент от **TComboBox**

Окно адреса Netscape Communicator, диалог "Выполнить" в Windows 98 и другие программы осуществляют очень дружелюбный интерфейс, известный обычно, как завершение текста. Этот документ описывает, как добавить подобные функциональные особенности в **TComboBox**.

Наиболее изящный способ и многократного использования добавить эти функциональные возможности - спуститься от **TComboBox** и *overriding* **ComboWndProc**, чтобы обратиться с сообщением

WM_KEYUP

. Добавляя новое свойство

TextCompletion

, функциональные возможности могут быть отключены, чтобы использовать

ComboBox

, подобно стандартному.

```
unit CompletingComboBox; interface uses Windows, Messages, SysUtils, Classes,
Graphics, Controls, Forms, Dialogs, StdCtrls; type TCompletingComboBox =
class(TComboBox) private FTextCompletion: Boolean; function GetTextCompletion:
Boolean; procedure SetTextCompletion(const Value: Boolean); protected // override the
WndProc() so that we can trap KeyUp events. procedure ComboWndProc(var Message:
TMessage; ComboWnd: HWnd; ComboProc: Pointer); override; public { Public
declarations } published property TextCompletion: Boolean read GetTextCompletion
write SetTextCompletion; end; procedure Register; implementation procedure
Register; begin RegisterComponents('Standard', [TCompletingComboBox]); end; {
TCompletingComboBox } function TCompletingComboBox.GetTextCompletion: Boolean;
begin Result := fTextCompletion; end; procedure
TCompletingComboBox.SetTextCompletion( const Value: Boolean); begin
fTextCompletion := Value; end; procedure TCompletingComboBox.ComboWndProc( var
Message: TMessage; ComboWnd: HWnd; ComboProc: Pointer); var rc, len: Integer;
begin inherited; case Message.Msg of WM_KEYUP: begin // test to see if its a
character that should not be // processed. if (Message.WParam 8) and
(Message.WParam VK_DELETE) and (Message.WParam VK_SHIFT) and
(FTextCompletion = True) then begin // Use CB_FINDSTRING to locate the string
in the Items // property rc := Perform(CB_FINDSTRING, -1,
Integer(PChar(Caption))); // if its in there then add the new string to the Text // and
```

```
select the portion that wasn't typed in by the // user if rc CB_ERR then
begin // store the length of the current string len := Length(Text); //
set the new string ItemIndex := rc; // highlight the rest of the text that was
added. SelStart := len; SelLength := Length(Text) - len; //
return 0 to signify that the message has been // handled. Message.Result := 0;
end; end; end; end; // case end; end.
```

По материалам <http://delphi.3000.com>