



Chapter 5

VCL Components

Professor Zhaoyun Sun





5.1 Visual Components

- Visual components include components such as edit controls, buttons, list boxes, labels, and so on.
- Most components you will use in a Delphi application are visual components.
- Visual components, as much as possible, show you at design time what the component will look like when the program runs.





5.1 Visual Components

- **New Term: Some components are visual components; others are nonvisual components. A visual component, as its name implies, is one that can be seen by the user at design time.**





5.2 IMPORTANT COMMON PROPERTIES

- All components have certain properties in common. For example, all visual components have Left and Top properties that determine where the component is placed on the form.





5.2 IMPORTANT COMMON PROPERTIES

- Properties such as Left, Top, Height, and Width are self-explanatory, so I won't go over them here. A few of the common properties, however, warrant a closer look.





The Align Property And The Color Property

- The Color property and The Align Property are simple to use ,we will not discuss them here.





5.2 IMPORTANT COMMON PROPERTIES

- ❑ The Cursor Property
- ❑ The Enabled Property
- ❑ The Font Property
- ❑ The Hint Property
- ❑ The ParentColor, ParentCtl3D, ParentFont, and ParentShowHint Properties





Other Common Properties

Additional Component Properties

<i>Property</i>	<i>Description</i>
BorderStyle	Can be <code>bsSingle</code> or <code>bsNone</code> . Use <code>bsNone</code> when you want the component to blend in with the background.
BoundsRect	The rectangle of the entire component (not limited to only the client area).
Caption	Sets the component's caption. Many components don't have captions, so for those components the <code>Caption</code> property is not exposed.
ClientHeight	Contains the height of the client area of the component.
ClientRect	Contains the rectangle for the client area of the component.
ClientWidth	Contains the width of the client area of the component.
Constraints	Sets the size constraints for the component (maximum width and height, minimum width and height). More important for forms than for other components. <code>Ctl3D</code> Indicates whether the control should be drawn with a 3D border. If <code>BorderStyle</code> is set to <code>bsNone</code> , this property has no effect.





Other Common Properties

Additional Component Properties

<i>Property</i>	<i>Description</i>
Height	Sets the component's height.
HelpContext	The HelpContext property is used to associate an index number in a help file with a particular component.
Left	Sets the x-coordinate of the component.
Parent	A pointer to the parent of the component.
PopupMenu	Specifies the pop-up menu that will be displayed when the user clicks the secondary mouse button.
TabOrder	For windowed components. Sets this component's position in the tab order.
TabStop	For windowed components. Indicates that this component can be tabbed into. Setting this property to False removes the component from the tab order.
Top	Sets the y-coordinate of the component.
Visible	When read, indicates whether the component is currently visible. When written to, Visible either hides or shows the component.
Width	Sets the width of the component.





Primary Methods of Components

Common Methods Of Components

<i>Method</i>	<i>Description</i>
Broadcast	Used to send a message to all windowed child components.
ClientToScreen	Converts client window coordinates into screen coordinates.
ContainsControl	Returns True if the specified component is a child of the component or form.
HandleAllocated	Returns True if the Handle property for the component has been created. Simply reading the Handle property automatically creates a handle if it hasn't already been created, so HandleAllocated can be used to check for the existence of the handle without creating it.
Hide	Hides the component. The component is still available to be shown again later.
Invalidate	Requests that the component be redrawn. The component will be redrawn at Windows's earliest convenience.
Perform	Sends a message directly to a component rather than going through the Windows messaging system.
Refresh	Requests that a component be redrawn immediately and erases the component prior to repainting.
Repaint	Requests that a component be redrawn immediately. The component's background is not erased prior to repainting.
SetBounds	Enables you to set the Top, Left, Width, and Height properties all at one time. This saves time having to set them individually.
SetFocus	Sets the focus to a component and makes it the active component. Applies only to windowed components.
Update	Forces an immediate repaint of the control. Typically, you should use Refresh or Repaint to repaint components.





5.3 Common Events

- As with properties and methods, there are some events that will be responded to most often.
- Components cover a wide variety of possible Windows controls, so each component will have individual needs. The most commonly used events are listed in Table 5.3.





5.4 Common Events

Commonly Handled Component Events

<i>Event</i>	<i>Description</i>
OnChange	This event is triggered when a control changes in one way or another. Exact implementation depends on the component.
OnClick	Sent when the component is clicked with either mouse button.
OnDblClick	This event occurs when the user double-clicks the component.
OnEnter	This event occurs when a windowed component receives focus (is activated).
OnExit	This event occurs when a windowed component loses focus as the result of the user switching to a different control. It does not occur, however, when the user switches forms or switches to another application.
OnKeyDown	This event is triggered when the user presses a key while the control has focus. Keys include all alphanumeric keys as well as keys such as the arrow keys, Home, End, Ctrl, and so on.
OnKeyPress	This event is also triggered when the user presses a key, but only when alphanumeric keys or the Tab, backspace, Enter, or Esc keys are pressed.
OnKeyUp	This event occurs whenever a key is released.
OnMouseDown	This event is triggered when the mouse button is pressed while it's over the component. The parameters passed to the event handler give you information on which mouse button was clicked, special keys that were pressed (Alt, Shift, Ctrl), and the x,y coordinate of the mouse pointer when the event occurred.
OnMouseMove	This event occurs any time the mouse is moved over the control.
OnMouseUp	This event is triggered when the mouse button is released while over a control. The mouse button must first have been clicked while on the control.
OnPaint	This event is sent any time a component needs repainting. You can respond to this event to do any custom painting a component requires.





5.4 Edit Controls

- Delphi comes with four edit-control components. The Edit, Memo, and MaskEdit components are based on the standard Windows edit control.
- The RichEdit component is based on the Win32 rich edit control, which is not one of the standard Windows controls.





The MaskEdit Component

The Input Mask Editor

Input Mask: (99/99/00;1;_

Character for Blanks: _

Save Literal Characters

Test Input: ___-___

Masks... OK Cancel Help

Sample Mask	Sample Value
Phone	(415)555-1212
Extension	15450
Social Security	555-55-5555
Short Zip Code	90504
Long Zip Code	90504-0000
Date	06-27-94
Long Time	09:05:15PM
Short Time	13:45





5.4 Edit Controls

- ❑ The Memo Component
- ❑ The RichEdit Component





5.5 Common Edit Control Properties

Table 5.4. Properties For Edit Controls

Item	Applies To	Description	Properties
AutoSelect	Edit, MaskEdit	When set to True, text in the edit control will automatically be selected when the user tabs to the control. Default: True	
AutoSize	Edit, MaskEdit	When set to True, the edit control will automatically resize itself when the font of the edit control changes. erwise, the edit control does not change size when the font changes. Default: True	
CharCase	Edit, MaskEdit	Determines whether the edit control displays uppercase(ecUpperCase), lowercase (ecLowerCase), or mixed text (ecNormal). Default: ecNormal.	
HideScrollBars	RichEdit	When set to True, the scrollbars will be shown when needed but hidden otherwise. When set to False, the scrollbars are shown as determined by the value of the ScrollBars property.	
HideSelection	Edit, Memo, RichEdit	When set to True, any text selected will not show as selected when the user tabs to another control. Default: False	
Lines	Memo, RichEdit	The text contained in the component. Lines is an instance of the TStrings class.	
MaxLength	All	Specifies the maximum number of characters that the component will hold. When set to 0, the amount of text that can be input is unlimited (limited only by system considerations). When set to any non-zero value, limits the number of characters to that value. Default: 0	
OEMConvert	Edit, Memo	Set this property to True when the text input will consist of filenames. Default: False	
PasswordChar	Edit, MaskEdit	When this property is set to a value other than ASCII #0, any text entered will be echoed with the character provided. The actual text in the edit control is unaffected. Most password edits use the asterisk (*) as the password character. Default: #0	





5.5 Common Edit Control Properties

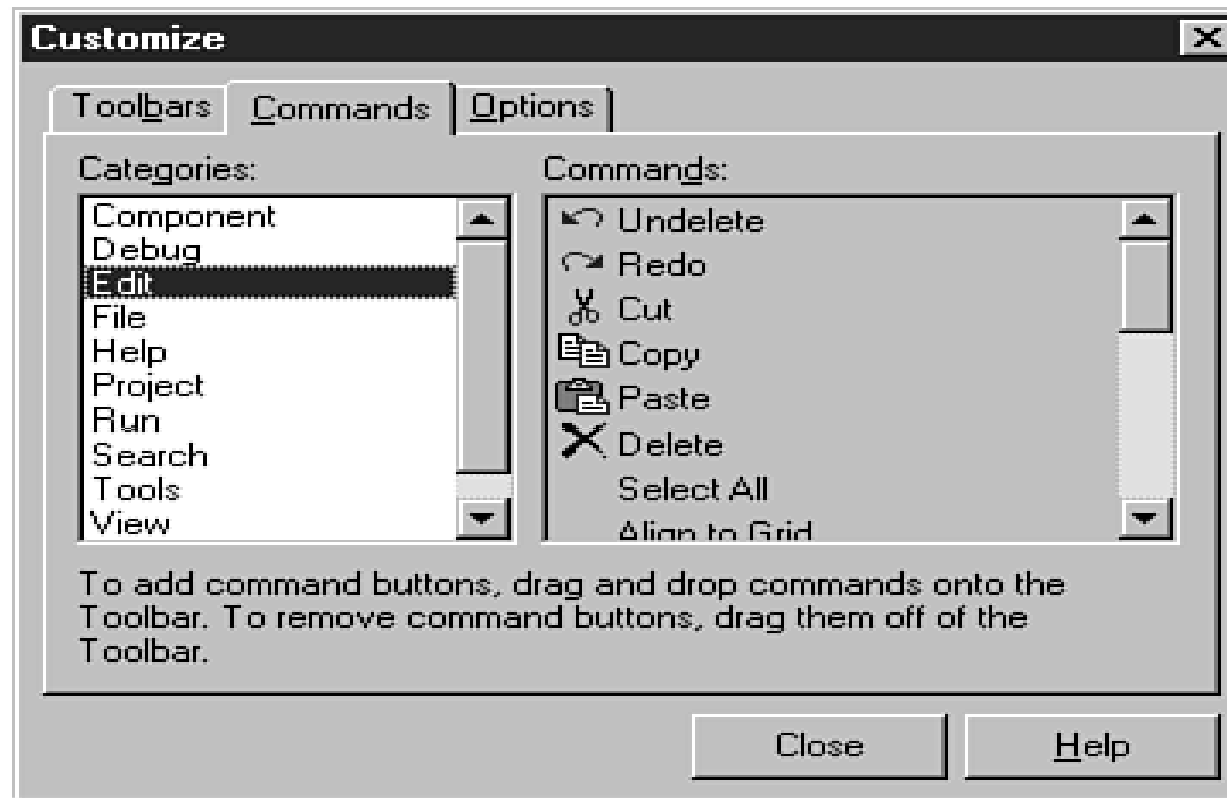
Item	Applies To	Description
Properties		
PlainText	RichEdit	When set to True, RTF (rich text format) files will be shown as plain text without character and paragraph formatting. When set to False, RTF files are displayed with full formatting. Default: False
ReadOnly	All	When set to True, the component will display its text, but new text cannot be entered. The user can, however, highlight text and copy it to the Clipboard. Default: False
ScrollBars	Memo, RichEdit	Determines which scrollbars to display. Choices are ssNone, ssBoth, and ssVertical. Default: ssNone
Text	Edit, MaskEdit	Contains the text in the component.
WantReturns	Memo, RichEdit	When set to True, the component keeps the return character and a new line is inserted in the edit control when the user presses Enter. When set to False, return characters go to the form and are not placed in the edit control. If you have a form with a default button and WantReturns set to False, pressing Enter will cause the form to close. Default: True
WantTabs	Memo, RichEdit	When set to True, a tab character is placed in the edit control when the user presses the Tab key. When set to False, tab characters, which would enable tabbing out of the edit control. Default: False
WordWrap	Memo, RichEdit	When set to True, text entered will wrap to a new line when the right edge of the edit control is reached. When set to False, the edit control automatically scrolls as new text is entered. Default: True
Modified	All	Indicates whether the contents of the edit control have changed since the last time the Modified property was set. After saving the contents of a Memo to a file, you should set Modified to False.
SelLength	All	Contains the length of the text currently selected in the edit control.
SelStart	All	Contains the starting point of the selected text in the edit control. character in the edit control is 0.
SelText	All	Contains the currently selected text in an edit control.





5.6 The ListBox and ComboBox Components

The Toolbar Editor's Commands List Box Is Owner-Drawn





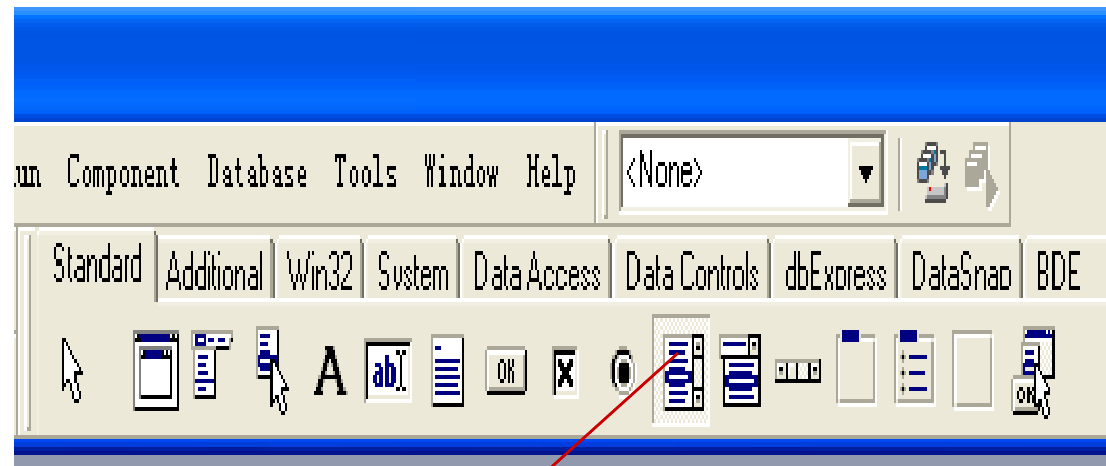
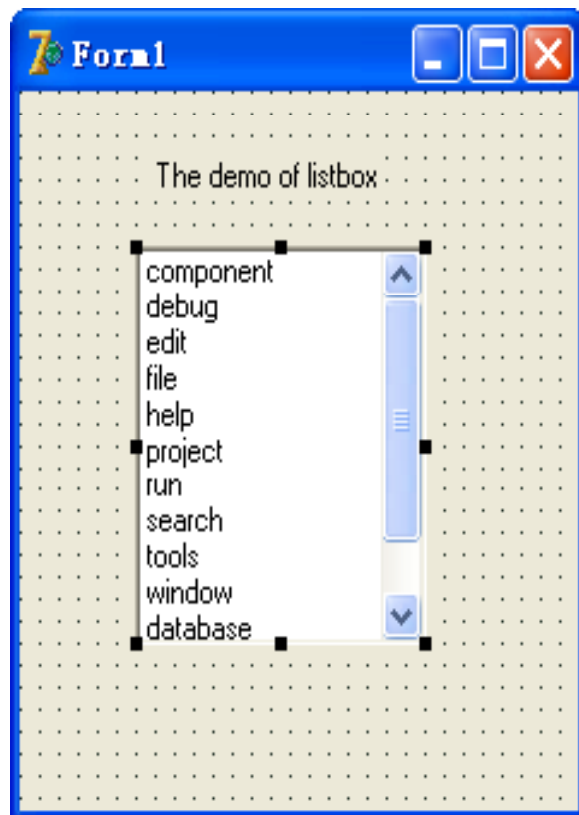
5.6 The ListBox and ComboBox Components

- The ListBox and ComboBox components are also widely used. The ListBox component represents a standard Windows list box, which simply presents a list of choices that the user can choose from.**





5.6 The ListBox and ComboBox Components



Listbox component





5.6 The ListBox and ComboBox Components

- If the list box contains more items than can be shown at one time, scrollbars provide access to the rest of the items in the list box.**





5.6 The ListBox and ComboBox Components

- The list box on the left is a regular list box; it lists the possible button groups you can choose from. The list box on the right is an owner-drawn list box.**





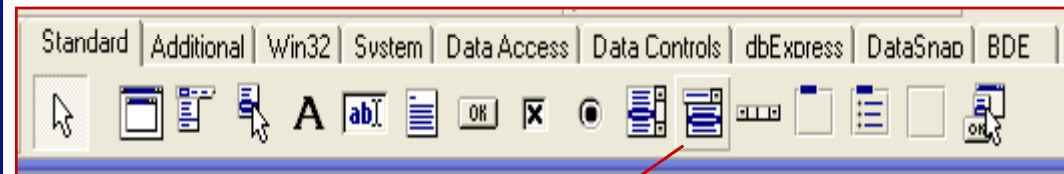
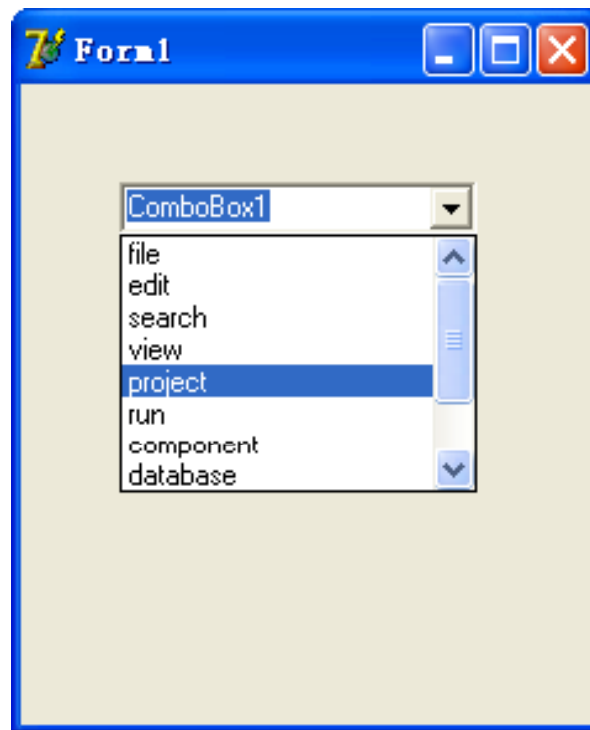
5.6 The ListBox and ComboBox Components

- It shows the actual button as it will appear on the toolbar, as well as a textual description of what function the button performs. Combo boxes are specialized list boxes. Actually, a combo box is a combination of a list box and an edit control.**





5.6 The ListBox and ComboBox Components



Combobox component





5.6 The ListBox and ComboBox Components

- ❑ The user can choose from the list or type in a value in the edit portion. When the user chooses an item from the list, that item is placed in the edit control.**





5.6 The ListBox and ComboBox Components

- There are three different types of combo box. The combo box type is determined by the Style property. Table 5.5 lists the types of combo boxes and a description of each.**





5.6 The ListBox and ComboBox Components

Types Of Combo Boxes

<i>Property</i>	<i>Applies To</i>	<i>Description</i>
Properties		
MaxLength	ComboBox	The maximum number of characters the user can type in the edit portion of the combo box. Same as MaxLength in edit controls. Default: 0
MultiSelect	(no limit) ListBox	When True, the list box enables multiple items to be selected. Default: False
Sorted	Both	When set to True, the list box items are sorted in ascending order. When set to False, the items are not sorted. Default: False
Style	ComboBox	The style of the combo box. Choices are csSimple, csDropDown, csDropDownList, lbOwnerDrawFixed, and csOwnerDrawVariable. Default: csDropDown
	ListBox	Style choices for list boxes are lbStandard, lbOwnerDrawFixed, and csOwnerDrawVariable. Default: lbStandard
TabWidth	ListBox	List boxes can use tabs. This property sets the tab width in pixels. Default: 0





5.6 The ListBox and ComboBox Components

Properties For Edit Controls

<i>Property</i>	<i>Applies To</i>	<i>Description</i>
Properties		
Columns	ListBox	Contains the number of columns in the list box. You can create multiple columns by making this property greater than 1. Default: 0
ExtendedSelection	ListBox	Determines whether extended selection is allowed. <i>Extended selection</i> enables the user to select items using Shift+click and Ctrl+click. Has no effect if MultiSelect is set to False. Default: True
IntegralHeight	ListBox	When True, the list box height will be resized to be sure that no partial lines are displayed. When False, the list box might show partial lines. Default: False
ItemHeight	Both	For use with owner-drawn list boxes and combo boxes. Sets the height of the items in the control. Default: 13
Items	Both	A TStrings instance that contains the list of items in the list box.(See the section on TStrings earlier in this chapter for a description of available properties and methods.)

