

select *Ease of Use* and *Change how your keyboard works*. Then, in either, check the box for *Underline keyboard shortcuts and access keys*.

## Setting the Accept and Cancel Buttons

Are you a keyboard user? If so, do you mind having to pick up the mouse and click a button after typing text into a text box? Once a person's fingers are on the keyboard, most people prefer to press the Enter key, rather than to click the mouse. If one of the buttons on the form is the Accept button, pressing Enter is the same as clicking the button.

You can make one of your buttons the Accept button by setting the **AcceptButton** property of the form to the button name. When the user presses the Enter key, that button is automatically selected.

You also can select a *Cancel button*. The Cancel button is the button that is selected when the user presses the Esc key. You can make a button the Cancel button by setting the form's **CancelButton** property. An example of a good time to set the CancelButton property is on a form with *OK* and *Cancel* buttons. You may want to set the form's AcceptButton to OKButton and the CancelButton property to CancelButton.

## Setting the Tab Order for Controls

In Windows programs, one control on the form always has the **focus**. You can see the focus change as you tab from control to control. For many controls, such as buttons, the focus appears as a thick border. Other controls indicate the focus by a dotted line or a shaded background. For text boxes, the insertion point (also called the *cursor*) appears inside the box.

Some controls can receive the focus; others cannot. For example, text boxes and buttons can receive the focus, but labels and picture boxes cannot.

### The Tab Order

Two properties determine whether the focus stops on a control and the order in which the focus moves. Controls that are capable of receiving focus have a **TabStop** property, which you can set to True or False. If you do not want the focus to stop on a control when the user presses the Tab key, set the TabStop property to False.

The **TabIndex** property determines the order the focus moves as the Tab key is pressed. As you create controls on your form, Visual Studio assigns the TabIndex property in sequence. Most of the time that order is correct, but if you want to tab in some other sequence or if you add controls later, you will need to modify the TabIndex properties of your controls.

When your program begins running, the focus is on the control with the lowest TabIndex (usually 0). Since you generally want the insertion point to appear in the first control on the form, its TabIndex should be set to 0. The next control should be set to 1; the next to 2; and so forth.

You may be puzzled by the properties of labels, which have a TabIndex property but not a TabStop. A label cannot receive focus, but it has a location in the tab sequence. This fact allows you to create keyboard access keys for text boxes. When the user types an access key that is in a label, such as Alt + N, the focus jumps to the first TabIndex following the label (the text box). See Figure 2.17.

By default, buttons, text boxes, and radio buttons have their TabStop property set to True. Be aware that the behavior of radio buttons in the tab sequence