

Using Custom Mouse Cursors

Most Visual Basic programmers know that you can use an object's (a control or form) `MousePointer` property to change the appearance of the mouse pointer when it is over that object. As useful as this can be, it limits you to selecting from Visual Basic's predefined pointers. It's true that this set of predefined pointers includes those that are needed most often, such as the hourglass, crosshairs, and I-beam. But what if you need a mouse pointer that is not provided in this set? This tip shows you how to use any icon as your mouse pointer.

Icons are small images that are saved in ICO files. They are typically 32x32 pixels in size. You can create your own icons using one of the many shareware icon editors that are available, or you can use any of the icons that are provided as part of the Visual Basic installation (by default in the `\Program Files\Microsoft Visual Studio\Common\Graphics\Icons` folder).

Once you know the icon you want to use, there are two steps required. The first is to assign the icon to the object's `MouseIcon` property, using the `LoadPicture` function to load the icon file. Then, set the object's `MousePointer` property to the value 99. For example, assume that your project contains a picture box control named `Picture1`. Putting this code in the form's `Load` event procedure will cause the specified icon to be displayed when the mouse pointer is over the picture box. This code assumes that the icon file is located in the folder from which the program is running.

```
Picture1.MouseIcon = LoadPicture("arw03dn.ico")  
Picture1.MousePointer = 99
```

You should be aware that each icon has a "hotspot" associated with it. This spot identifies the precise spot that is the mouse pointer's location in terms of clicking and other operations. For example, the hotspot of the standard arrow cursor is at the tip of the arrow. If you create your own cursors using an icon editor you can define the location of the hotspot as well.