



The Incredible Scrolling Credits Box Contents

[About The Incredible Scrolling Credits Box](#)

[Changing The Background Color](#)

[Changing The Color Of The Bouncing Balls](#)

[Putting Your Text Into The Credits Box](#)

[Making The Credits Scroll Faster](#)

[How To Increase The Number Of Bouncing Balls](#)

[Making The Credits Box Come Alive](#)

The Incredible Scrolling Credits Box

About This Component And Its Author

After looking at all of the interesting components that are shipped with Delphi, I discovered a disturbing lack. There was nothing available that let you scroll text smoothly in a vertical direction. Actually, there isn't anything built in that lets you scroll text in any direction. So, to rectify this situation, I decided that it would be really neat to have some sort of vertical scrolling text box that would be useful for displaying credits or other information. And so, from this inspiration, the scrolling credits box was born. It's fairly easy to use. All you need to do is drop it on your form in the size that you want, choose a color for the background, bouncing balls, and text, select your font, type in the text you want displayed, choose the number of balls you want bouncing around, and set Started to True. You can view this at design time, but I would suggest leaving Started set to False until you display it in the program. It does hook onto one of the system timers, and this may cause a lack of response from Windows as it is continually updating the component.

An OnClick and OnDoubleClick event can be specified so that the component can be used as some sort of control, but its main purpose is just a nifty little scrolling text box. I hope you find this component at least mildly entertaining, if not somewhat useful. I enjoyed writing it, and I hope you enjoy using it. If you would like to use this component in your programs, I am only asking for \$5 as a registration fee, and perhaps a small mentioning in your credits, or maybe just a copy of what you've done with it. If you would like the source code, which comes complete with comments and the source for this help file, I am asking \$25. Please send your registration, comments, or suggestions for improvement to:

John Ayres
P.O. Box 294903
Lewisville, TX 75029-4903

Background Property

The Background property allows you to change the color of the background behind the bouncing balls and scrolling text. It defaults to black. On screens with a high resolution, you will want to make sure that the colors of the balls, background, and text are not very similar, as the colors may blend, and the balls especially might be very hard to see.

Ballcolor Property

The Ballcolor property allows you to change the color of the bouncing balls. It defaults to white. On screens with a high resolution, you will want to make sure that the colors of the balls, background, and text are not very similar, as the colors may blend, and the balls especially might be very hard to see.

Credits Property

The Credits property is the list of text that you want to appear in the credits box. The credits box will automatically center your text as it is scrolling. However, if the text line is long, it may not fit entirely in the credits box, and will be clipped on either side. You can expand the credits box or use a smaller font to try and fit it in.

Increment Property

The Increment property lets you control how fast the text scrolls up the credits box. You can set this number from 1 to 5, and the higher the number, the faster it will scroll.

NumberBalls Property

The NumberBalls property simply sets the number of balls that will be seen bouncing around in the box and behind the credits. This number can be from 0 to 10, and can only be set if Started is set to False. The more balls there are, the slower the component will appear to run. However, this effect is usually not very noticeable, especially on faster machines.

Started Property

When set to True, the credits box will come alive and you will see your text scrolling slowly up inside the box with small balls bouncing around behind the credits. When you set this to False, and then set it to True again, it resets the location of your text so it starts scrolling from the bottom again. This component hooks into the system timer, and with the Started property set to True at design time, you may notice a lack of system response. Typically, this manifests itself as an inability to hold the mouse button down on a down arrow of a list box (like the box for selecting the name of a font, for example). You can scroll down through a list box by repeatedly pressing the mouse buttons, though. Generally, it is useful to set Started to True during design time only to see what it will look like with your text. Set it to false while designing your program, and then set it to True programmatically when you display the credits box.

