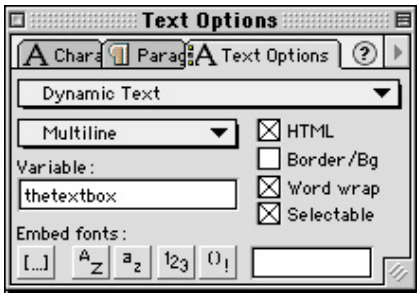
Scrolling text boxes in Flash

Setting things up

Okay, first of all you'll need to make two buttons: up and down. They can be of any shape you want.

Next you'll need to make a textbox. Just click on the text tool and let your inspiration carry you. You can let the text overflow we'll fix this later.

Now select the text and go into the « Text Options » panel. Set the options as in this screenshot:

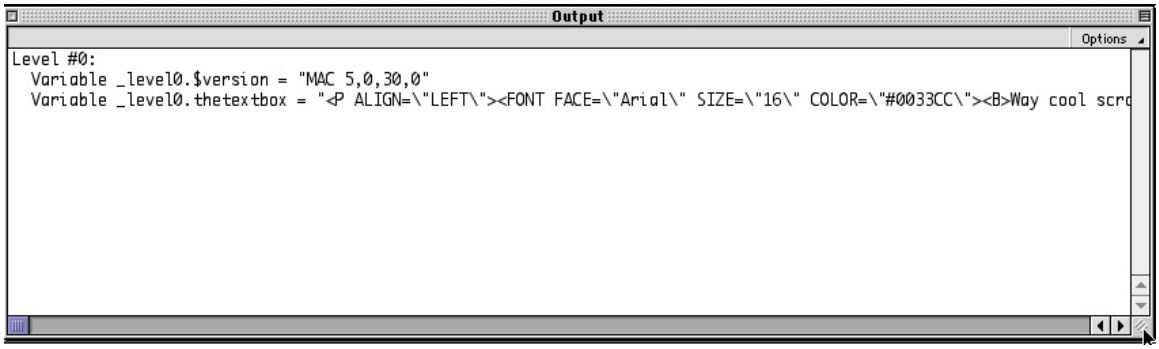


If you can't seem to find the panel, it's in Window> Panels> Text options. Here's what each of the options do:

1. We choose dynamic text, since it is the only kind of textbox where you can alter the scroll property in actionscript.
2. Choosing Multiline will allow the text to spread across multiple lines.
3. We set the variable name to "thetextbox" here. We'll use this name later to refer to the textbox in ActionScript.
4. By choosing HTML, we can apply basic formatting on the text (color, size, bold, italics, and hyperlinks)
5. Word wrap works in conjunction with multiline to allow the text to wrap.

There is something peculiar in the way Flash handles text boxes and scrolling: **you can't scroll a text box unless you fill its contents using ActionScript**. We're just going to place the contents of the text box directly in Actionscript (a technique called «hard-coding»).

There's an easy way to transfer the text (or rather, the underlying HTML) from the text box to ActionScript. When you test your movie (Apple +enter), you can see a list of the variables in the movie by going to the Debug menu and choosing « List Variables ». You should see something similar to this:



As you can see, the HTML content of the text box is accessible (it's on the last line, in this case). Select all of the HTML, copy it, and then paste it into Simpletext or a similar program. Save this for later use.

Once this is done, you can go back to your Flash movie, and safely delete the contents of the text box. You should reduce its **height** as well otherwise there won't be a need for scrolling.

Create a new movie clip symbol (Apple + F8) and cut and paste your text box onto this. Place your two buttons on the main timeline on a new layer. Name it «container». Our setup is now done.

Adding the ActionScript

The first thing you'll want to do is load the text into the text box using ActionScript. Select container and go to the « Object Actions » panel. You can make this panel appear by pressing the arrow icon at the bottom right of the Flash window.

Now click on the arrow at the top right of the « Object Actions » panel, and select Expert Mode from the drop-down menu. This will allow you to edit the code directly, instead of having to « fill in the blanks ». Copy and paste this script inside of the window:

```
onClipEvent (load){
 thetextbox = "Insert text here";
 scrolling = 0;
 frameCounter = 1;
 speedFactor = 3;
}
onClipEvent (enterFrame){
 if( frameCounter % speedFactor == 0){
 if( scrolling == "up" && thetextbox.scroll > 1){
 thetextbox.scroll--;
 }
 else if( scrolling == "down" && thetextbox.scroll < thetextbox.maxscroll){
 thetextbox.scroll++;
 }
 frameCounter = 0;
 }
 frameCounter++;
}
```

You'll need to replace « Insert text here » with your own text (which you pasted into Simpletext, use the text from <P ALIGN onwards).

Note the use of onClipEvent(load) here. Everything that is placed inside a load clip event will be executed only once, when the clip is loaded.

Now, we want the scrolling to occur at regular intervals, a good way to achieve this is keeping a counter containing the number of the frame we're on. If that counter can be divided by a certain number, then you should scroll, otherwise not.

To check if a number is divisible by another one, we can use the modulo (%) operator. This gives us the remainder of a division. For example, 11 % 3 = 2, since 11 divided by 3 is 3, with 2 remaining. Knowing this, if frameNumber % skipFactor is equal to 0, then we should scroll, otherwise not.

However, when we first click on a button, we should be given immediate feedback regardless of what frame we're on. By setting the frameNumber to skipFactor when a button is clicked, the next time the onClipEvent(enterFrame) code is fired, the text box will scroll immediately. This is not a simple caprice: fast response times are vital to user experience.

Now that we know where we should scroll, it's time to actually do the scrolling. Dynamic text boxes have two properties: scroll and maxscroll. Each of the text lines is numbered, starting with 1. The scroll property is the number of the first line that is shown; you can change its value to get a scrolling effect. Maxscroll is the maximum value of scroll, that is, the total number of lines minus the number of lines shown at once; this one can't be set, just read.

Now that this is done, we have to figure out a way to keep track of whether we should scroll the text up or down, or do nothing at all. So we'll add these actions to the buttons:

Up button:

```
on( press ){
 scrolling = "up";
 frameCounter = speedFactor;
}
on( release, releaseOutside ){
 scrolling = 0;
}
```

Down button:

```
on( press ){
 scrolling = "down";
 frameCounter = speedFactor;
}
on( release, releaseOutside ){
 scrolling = 0;
}
```

So what's happening here is pretty simple: when one of the buttons is pressed, the variable «scrolling» is set to "up" or "down", depending on the button. When the button is released, the variable is set to 0, which is the programmatic equivalent of "no".

If you're not familiar with the use of ++ and --, here's a brief explanation. ++ adds 1 to the value of the variable or property it affects; this is known as incrementing the value. --, on the other hand, subtracts 1 from the value of a variable; this is called decrementing a value. If you test your movie now, you should see the scroller work!