



## Text: How an application handles characters, words, text and HTML

Revolution makes it very easy to work with text. Revolution understands text in the same way that you do – referring to it using words, lines, and characters. This tutorial takes you through the basics of writing scripts to work with text.

### Key topics covered in this tutorial

- You can work with text using natural, English-like statements
- How to refer to text in a script
- How to do a word, line or character count
- How to insert text into a paragraph
- How to work with the clipboard from a script
- Working with styled text using HTML

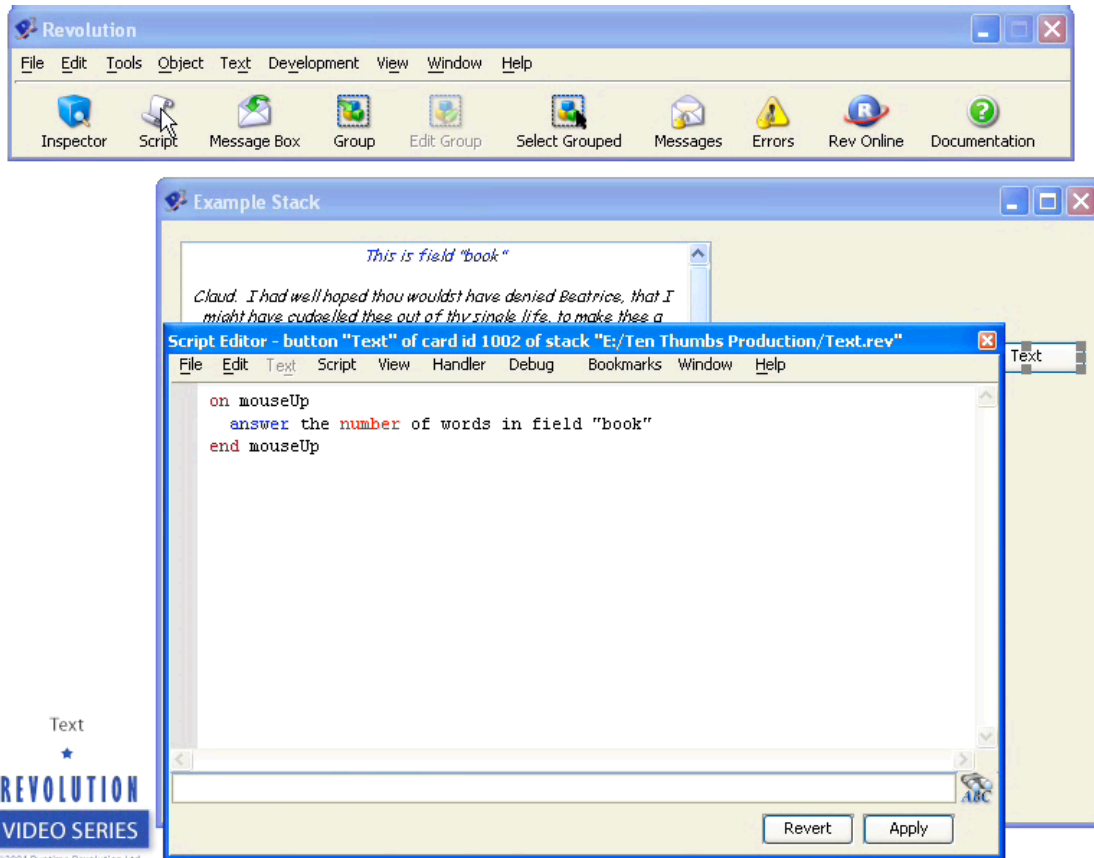
**See also:** [Reference documentation: Chunk expressions](#)

---

Revolution has a complete set of built-in text handling options. Revolution understands the concept of a word, character, or line in the same way that you do.

Let's start by trying a simple word count. To do this we'll run the following script in our button (we've already created the field with text in it and named it "book"):

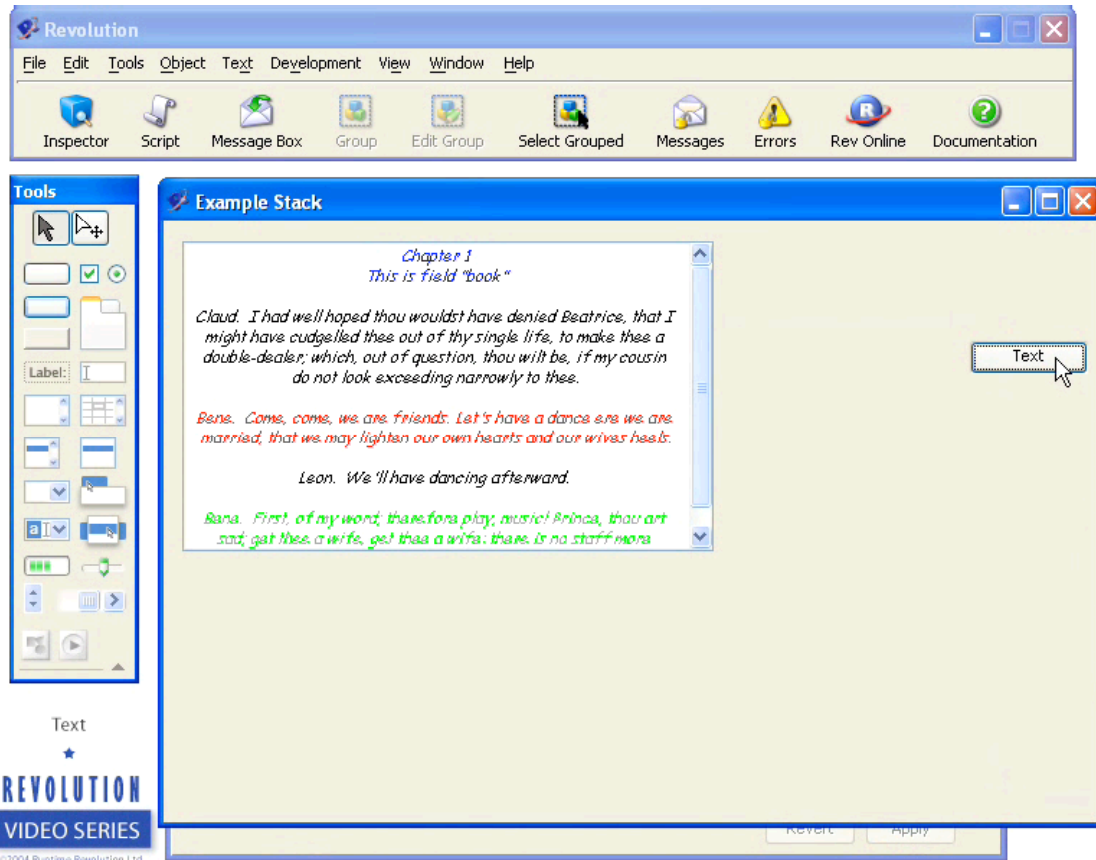
```
answer the number of words in field "book"
```



As you can see, Revolution understands the concept of a word. We can also do a character count by changing 'words' to 'characters'

Let's say we want to insert some words before the start of this text. We'll put the words "Chapter 1" and a new line, or return character, before the text in field "book":

put "Chapter 1" & return before field "book"

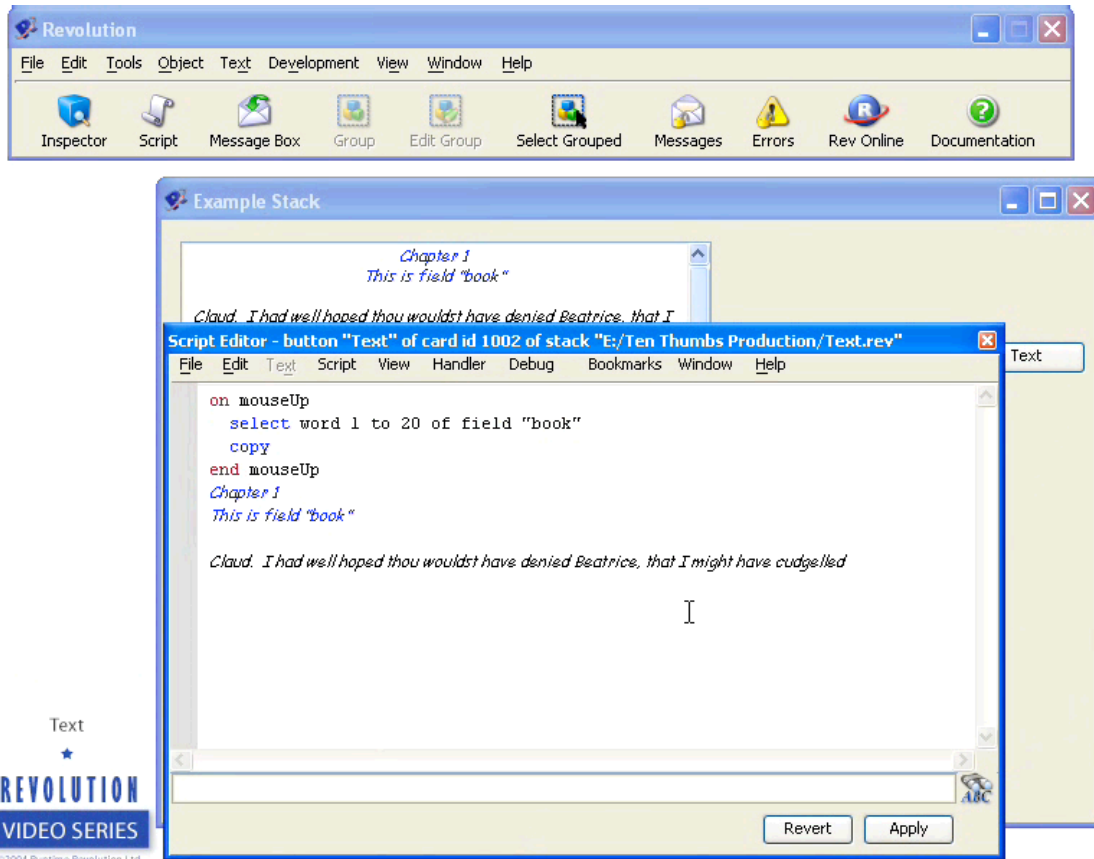


Now let's copy the first 20 words to the clipboard. First, we need to select the text, then copy it.

select word 1 to 20 of field "book"

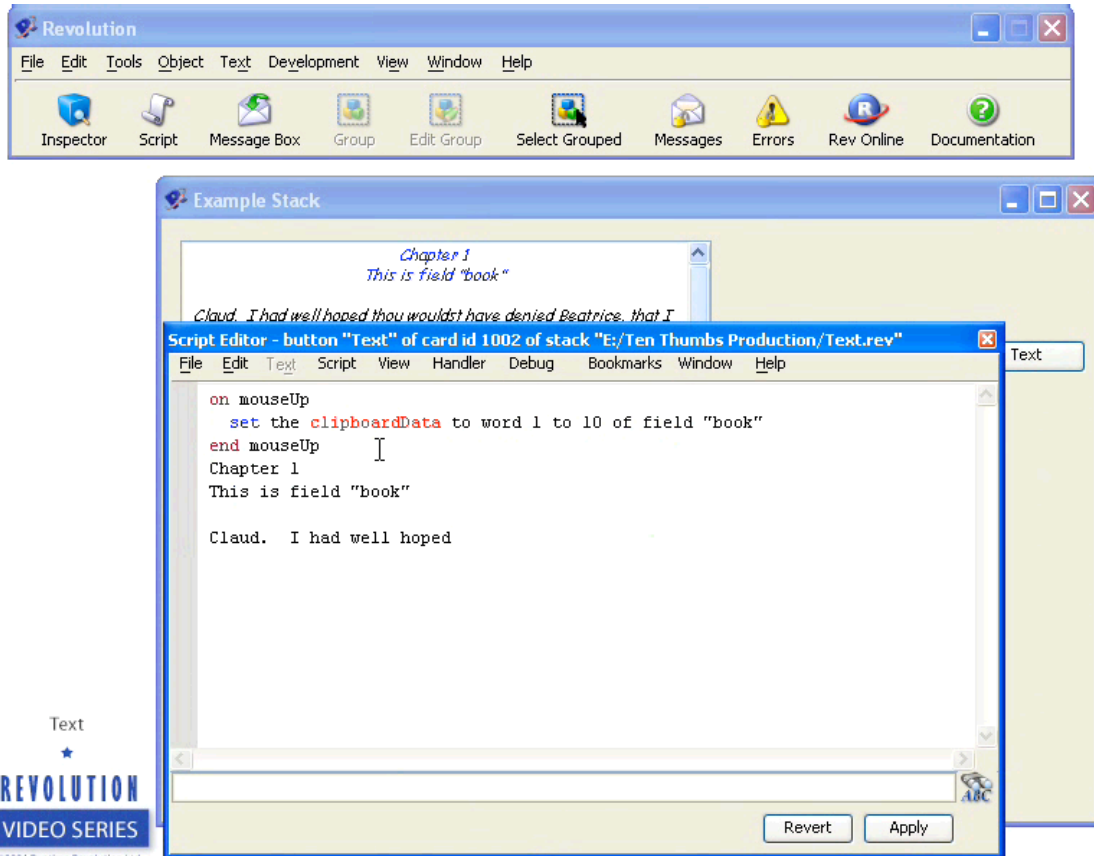
copy

We can then paste this text anywhere we want to (for this example we're using an unused part of the script editor for convenience).



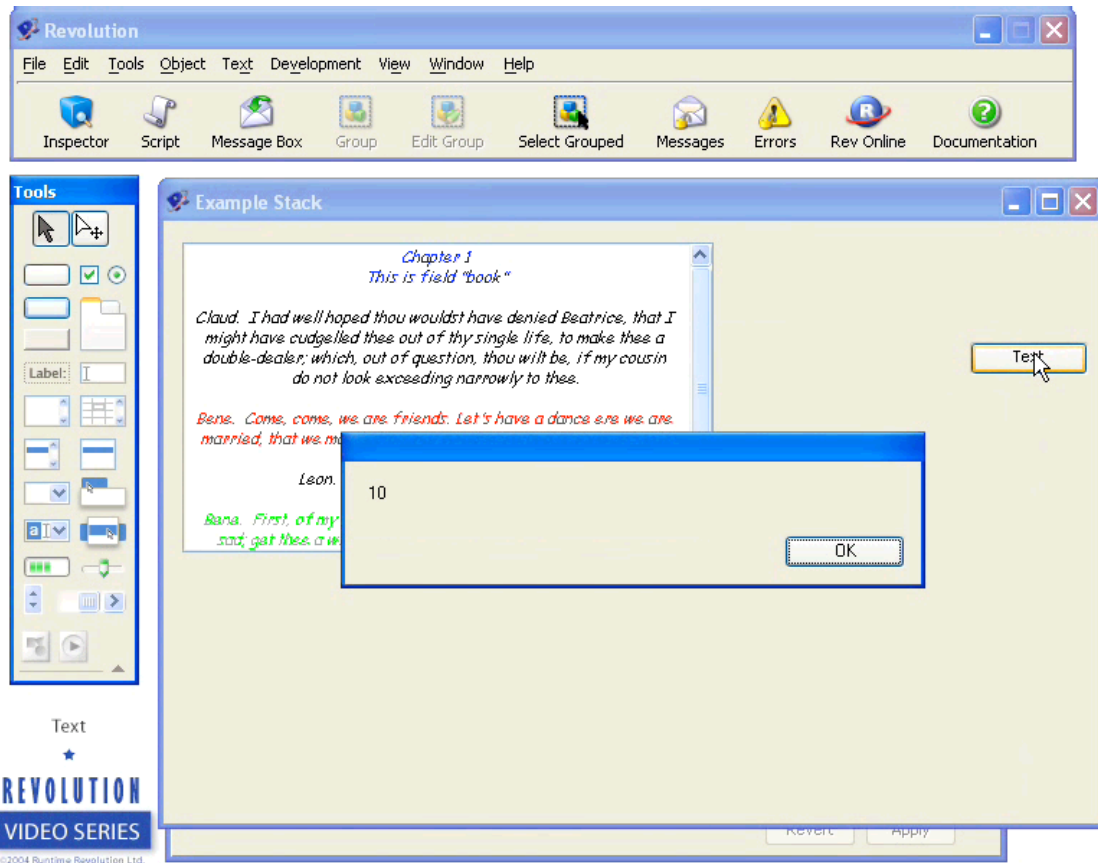
In fact, we don't need to select text to copy it, we can simply place the text directly onto the clipboard to have the same effect. The clipboard is a storage space in memory that all computers have that contains text, images or other information that has been copied, ready to paste.

set the clipboardData to word 1 to 10 of field "book"



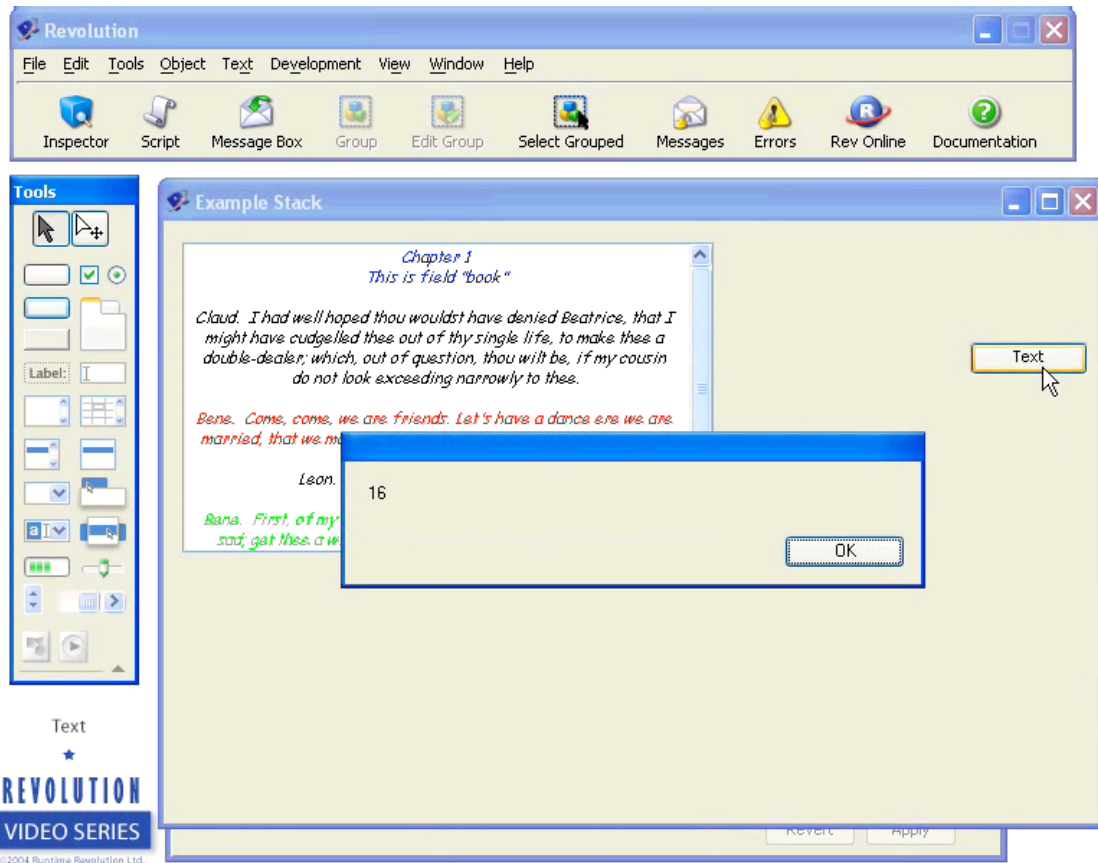
Let's say we want to do a line count on the text. There are two ways to do a line count – either we can count the number of lines based on the number of return characters in the text. Or we can count the number of actual lines displayed in the field given the current width of the field. Let's try the number of lines based on the number of return characters first.

answer the number of lines in field "book"



Now we can count the number of lines in the text as currently displayed in the field. To do that we use a field property called the 'formattedText', which calculates the text in the field as it is formatted to fit on the screen, effectively inserting the return characters that would be needed to make it stay wrapped the way it is.

answer the number of lines in the formattedText of field "book"



And you can see that there are considerably more lines reflecting the text wrapping. If we widen the field and run the script again you can see that the number of lines has gone down.

We can also work directly with text on the clipboard, without ever pasting it. For example, to do a word count of the text on the clipboard, we could:

answer the number of words in the clipboardData

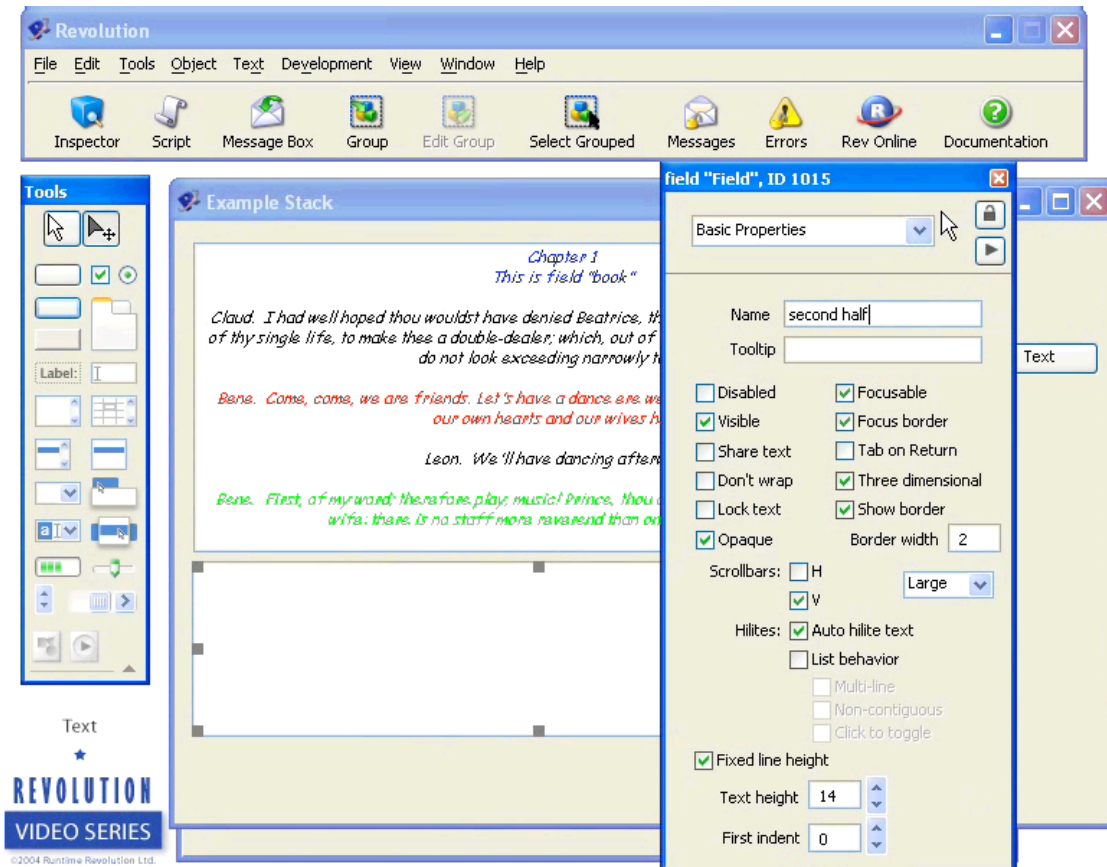
Now let's say we wanted to split our text in two. Let's leave half of the words in the field they are in, and move the other half to a new field. Firstly, we need to count the total number of words:

put the number of words in field "book" into tCount

Next we need to divide that number in half:

Divide tCount by 2

Now, let's create another field and we'll call this field 'second half'.



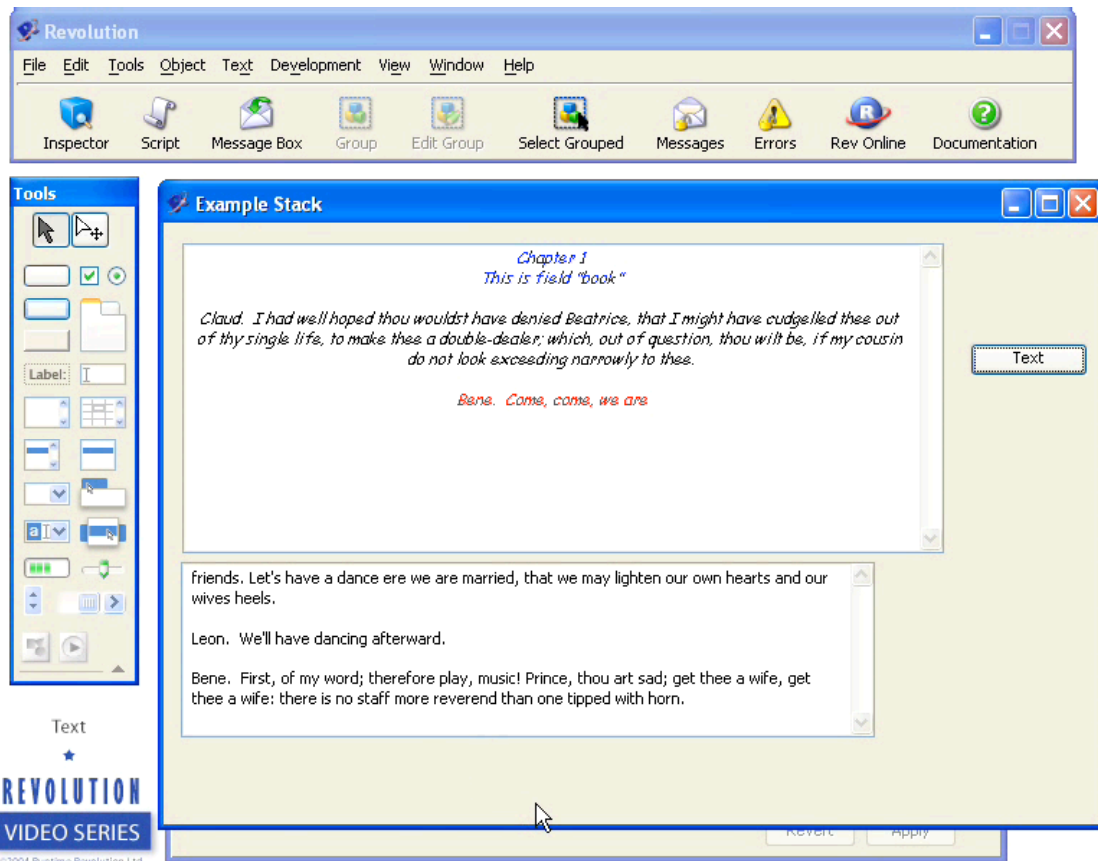


Now we're ready to copy the text between fields and remove the text from the first field:

```
put word tCount to -1 of field "book" into field "second half"
```

In this script we've used the put command to start half way through the text, the 'minus one' is our finishing point. If word 1 is the first word in the text; word -1 is the last word. If you count text using negative numbers, Revolution starts counting from the end of the text. So word -2 is the second last word, and so on. The last line of script we need to add is to remove the text now that we have copied it:

```
delete word tCount to -1 of field "book"
```

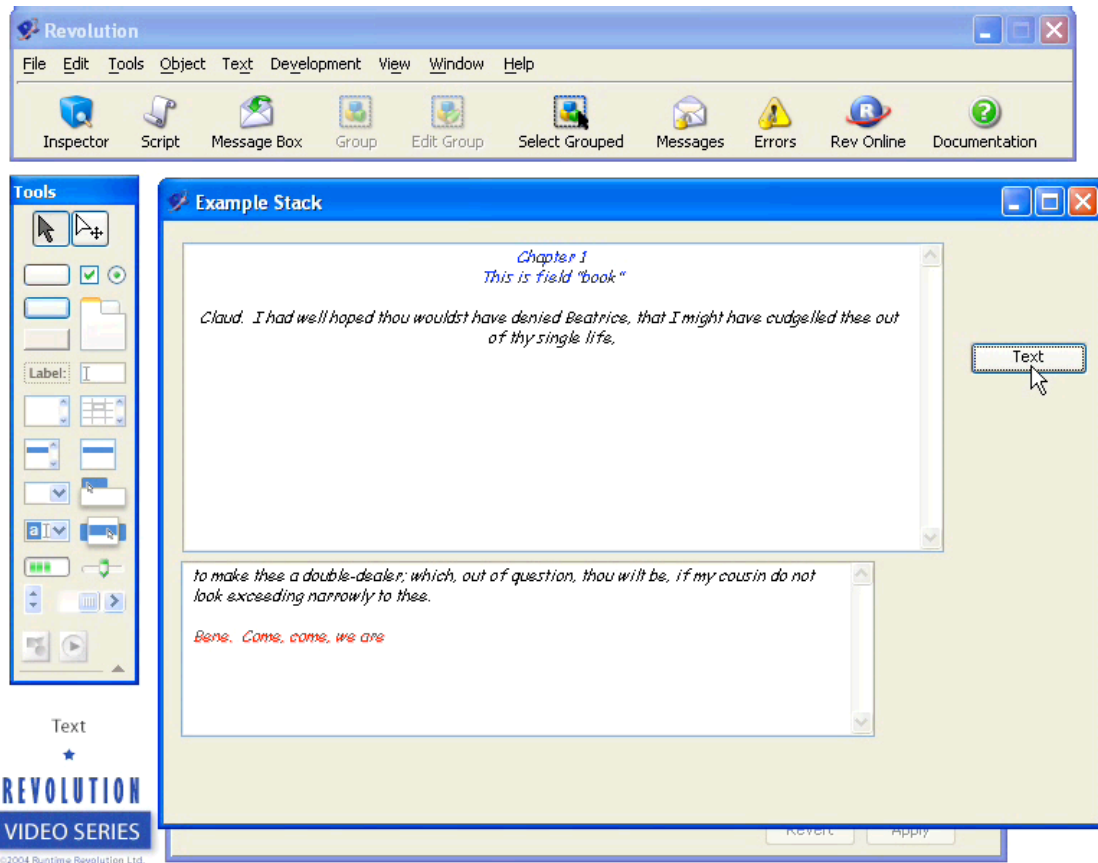


You'll notice right away that although the text has been copied, we've left the text styles behind. That's because the text property, which is what we have copied, refers to the raw text of a field, without formatting. This is useful to perform the calculations we were doing earlier, but isn't so useful when we want to maintain style. We could simply run a command to select the text and copy it, but there is another way to preserve styles. We can ask Revolution to give us the HTML equivalent of the text in the field. HTML is text that has been marked up to contain more information about the text, including styles.

Let's modify our script slightly. We'll change the third line of our script to:

```
set the htmlText of field "second half" to the htmlText of word tCount  
to -1 of field "book"
```

(This should be typed in a single line in the script editor.)



And there we see that half of the remaining text has been copied, complete with text styles.

We explore more advanced text manipulation features in the tutorial on importing text and advanced text handling.