

The [2023 Developer Survey results](#) are live!

## Current line and column numbers in a RichTextBox in a Winforms application

Asked 13 years, 3 months ago Modified 11 years, 10 months ago Viewed 20k times



How do I get the current line and column numbers in a RichTextBox in a Winforms application?

5

**NOTE**



Folks, I just want a simple solution, if it's available by someone and he's willing to share it with us, and not links to do research! Just give me some code please! Otherwise, I'll have to buy a control...



[c#](#) [.net](#) [vb.net](#) [winforms](#) [richtextbox](#)

Share Improve this question Follow

edited Mar 11, 2010 at 15:26

asked Mar 11, 2010 at 14:45  
user283622

3 Answers

Sorted by:

Highest score (default)



Thought I'd post a slightly simpler way of doing it.

18

```
// Get the line.  
int index = richTextBox.SelectionStart;  
int line = richTextBox.GetLineFromCharIndex(index);  
  
// Get the column.  
int firstChar = richTextBox.GetFirstCharIndexFromLine(line);  
int column = index - firstChar;
```



Get the current selected index, get the current line, then to get the column subtract the selected index from the index of the first character of that line.

Share Improve this answer Follow

answered Jul 29, 2011 at 16:36



[Alex McBride](#)

6,871 3 29 30

**Join Stack Overflow** to find the best answer to your technical question, help others answer theirs.

Sign up



For row and column, check out the [Richer RichTextBox](#) project. It is an extended version of the RichTextBox which supports the row and column numbers. The code from the article:

```

this.rtb.CursorPositionChanged +=
    new System.EventHandler(this.rtb_CursorPositionChanged);
this.rtb.SelectionChanged +=
    new System.EventHandler(this.rtb_SelectionChanged);
.
.
.
private void rtb_CursorPositionChanged(object sender, System.EventArgs e)
{
    int line = rtb.CurrentLine;
    int col = rtb.CurrentColumn;
    int pos = rtb.CurrentPosition;

    statusBar.Text = "Line " + line + ", Col " + col +
        ", Position " + pos;
}

private void rtb_SelectionChanged(object sender, System.EventArgs e)
{
    int start = rtb.SelectionStart;
    int end = rtb.SelectionEnd;
    int length = rtb.SelectionLength;

    statusBar.Text = "Start " + start + ", End " + end +
        ", Length " + length;
}

```

To achieve such behavior, we need to extend the RichTextBox class in the following manner:

```

using System;
using System.Drawing;
using System.Windows.Forms;

namespace Nik.UserControls
{
    public class RicherTextBox2 : System.Windows.Forms.RichTextBox
    {
        public event EventHandler CursorPositionChanged;

        protected virtual void OnCursorPositionChanged( EventArgs e )
        {
            if ( CursorPositionChanged != null )
                CursorPositionChanged( this, e );
        }

        protected override void OnSelectionChanged( EventArgs e )
        {
            if ( SelectionLength == 0 )
                OnCursorPositionChanged( e );
            else
                base.OnSelectionChanged( e );
        }
    }
}

```

```
{
    get { return CursorPosition.Column( this, SelectionStart ); }
}

public int CurrentLine
{
    get { return CursorPosition.Line( this, SelectionStart ); }
}

public int CurrentPosition
{
    get { return this.SelectionStart; }
}

public int SelectionEnd
{
    get { return SelectionStart + SelectionLength; }
}
}

internal class CursorPosition
{
    [System.Runtime.InteropServices.DllImport("user32")]
    public static extern int GetCaretPos(ref Point lpPoint);

    private static int GetCorrection(RichTextBox e, int index)
    {
        Point pt1 = Point.Empty;
        GetCaretPos(ref pt1);
        Point pt2 = e.GetPositionFromCharIndex(index);

        if ( pt1 != pt2 )
            return 1;
        else
            return 0;
    }

    public static int Line( RichTextBox e, int index )
    {
        int correction = GetCorrection( e, index );
        return e.GetLineFromCharIndex( index ) - correction + 1;
    }

    public static int Column( RichTextBox e, int index1 )
    {
        int correction = GetCorrection( e, index1 );
        Point p = e.GetPositionFromCharIndex( index1 - correction );

        if ( p.X == 1 )
            return 1;

        p.X = 0;
        int index2 = e.GetCharIndexFromPosition( p );

        int col = index1 - index2 + 1;

        return col;
    }
}
```

[Displaying line number in RichTextBox](#) for the line number portion.

Share Improve this answer Follow

edited May 23, 2017 at 12:02

answered Mar 11, 2010 at 14:53



Community Bot

1 1



SwDevMan81

48.7k 22 151 184

2 I don't know if it works, buddy, but you deserve to be voted. – user283622 Mar 11, 2010 at 17:12

1 Over complicating things when unnecessary. Just use a little thing called a Math to calculate positions (see Alex McBride answer). – SoLaR Feb 21, 2017 at 8:14



-1

[Line Numbers for RichText Control in C#](#)

[Numbering lines of RichTextBox in .NET 2.0](#)



Share Improve this answer Follow

answered Mar 11, 2010 at 14:48



sashaeve

9,357 10 48 61

