

```
58 //Nous assurons la position actuelle dans le flux
59 dataStream.Seek(firstOffset = firstLine * lineLength,
    SeekOrigin.Begin);
60
61 //Fond général
62 g.FillRectangle(backBr, r);
63
64 //Colonne des décalages
65 g.FillRectangle(offsetBackBr, new Rectangle(r.Location,
66     new Size(offsetZoneSz.Width, r.Height)));
67
68 //Marge du milieu
69 g.FillRectangle(middleMarginBr, new Rectangle(new Point(leftCurX
    + textZoneWidth, r.Y),
70     new Size(middleMarginWidth, r.Height)));
71
72 //Zones pour l'interaction avec la souris
73 leftZone = new Rectangle(leftCurX, r.Y, textZoneWidth,
    r.Height);
74 rightZone = new Rectangle(rightCurX, r.Y, textZoneWidth,
    r.Height);
75 offsetZone = new Rectangle(offsetCurX, r.Y, offsetZoneSz.Width,
    r.Height);
76
77 //todo gros tampon puis parcours (besoin de calculer le total
    d'abord. Mettre le tampon dans la classe générale)
78
79 //byte[] buf = new byte[lineLength];
80 Rectangle blockRectL = new Rectangle(leftCurX, r.Y, blockW,
    offsetZoneSz.Height);
81 Rectangle blockRectR = new Rectangle(rightCurX, r.Y, blockW,
    offsetZoneSz.Height);
82 long p = 0; //Position actuelle de début de lecture (emplacement
    dans le flux de buffer[0])
83
84
85 GetSelectionRange(out long sBegin, out long sEnd);
86
87
88 //Longueur du bloc lu en cours
89 int L = Min(bufferSz - bufferSz % lineLength, bufferSz); //avoir
    un nombre entier
90
    lignes *au max* (il se peut qu'on lise moins)
91 int i = 0;
92
93 unsafe
94 {
95     fixed (byte* pBuf = buffer)
96     {
97
98         byte* line = pBuf;
99         //parcours successif des tampons pour parcourir
```

```

    l'ensemble du fichier dont nous avons besoin.
100     while (dataStream.Position < dataStream.Length)
101     {
102         //p: position du début du bloc lu dans buffer (p + L
        est la dernière position)
103         p = dataStream.Position;
104         L = dataStream.Read(buffer, 0, bufferSz);
105         //          v vvvvvvvvvvvv éviter d'avoir une
        ligne partiellement visible
106         while (blockRectL.Y + lineHeight <= r.Bottom && i <
        L)
107         {
108
109             DrawOffset(g, stringW, offsetCurX, blockRectL,
        p, sBegin); //offset
110
111             int l = Min(lineLength, L - i);
112
113             DrawLine(g, BaseLeft, line, l, blockRectL, p,
        textZoneWidth, sBegin, sEnd); //Partie gauche
114             DrawLine(g, BaseRight, line, l, blockRectR, p,
        textZoneWidth, sBegin, sEnd); //Partie droite
115             blockRectL.Y += offsetZoneSz.Height;
116             blockRectR.Y += offsetZoneSz.Height;
117
118             line += 1;
119             i += 1;
120             p += 1;
121         }
122     } //dataStream.Position < dataStream.Length
123
124
125 }
126
127
128
129
130 }
131
132
133 lastOffset = p + i - 1; //au dernier tour, i est incrémenté de
        la longueur de la ligne ou du reste des octets. Il vaut donc
134         //la position juste après le dernier
        octet dessiné dans le tableau tampon.
135
136 lineHeight = offsetZoneSz.Height;
137
138 SetScrollBarLength(
139     firstLine == 0
140     &&
141     p + i == dataStream.Length ?
142         0
143         : /* pas à la fin */ GetScrollTicks

```

```
    ());  
144  
145    //g.DrawString(offsetMouse.ToString(), Font, foreBr, ↗  
    PointToClient(MousePosition));  
146    //g.DrawString(scrollBar.Value.ToString() + " de " + ↗  
    scrollBar.Maximum.ToString(), Font, backBrSel, r.Location + ↗  
    new Size(5, 5));  
147  
148    //Dessin de la sélection  
149    GetRectanglesFromOffset(selectionStart, out Rectangle s1, out ↗  
    Rectangle s2);  
150    if (!s1.IsEmpty)  
151    {  
152        // sous-caret  
153        switch (selectedZone)  
154        {  
155            case SubZone.LeftPanel:  
156                DrawCur(g, BaseLeft, offsetZoneSz, charW, ↗  
                nOfDigitsL, s1);  
157                break;  
158            case SubZone.RightPanel:  
159                DrawCur(g, BaseRight, offsetZoneSz, charW, ↗  
                nOfDigitsR, s2);  
160                break;  
161            default:  
162                break;  
163        }  
164  
165        g.DrawRectangle(caretPen, s1);  
166        g.DrawRectangle(caretPen, s2);  
167    }  
168  
169    if (selectedZone == SubZone.OffsetMargin && curInputingOffset >= ↗  
    0)  
170    {  
171        DrawInputingOffset(g, new Rectangle(offsetZone.X, ↗  
        offsetZone.Y, offsetZone.Width, lineHeight), charW, ↗  
        stringW);  
172    }
```