Subject: Re: BufferPainter::Clear() optimization
Posted by mirek on Tue, 19 May 2020 07:14:34 GMT
View Forum Message <> Reply to Message

Yeah, there was another bug in it, I should test more before posting.

In retrospective, while the trick is nice, I do not think it is worth it. But if you wanted to experiment with this path, I have found the way how to extend / simplify this. The basic idea is

```
int nlen = -len;
t[1 & HIBYTE(nlen)] = c;
nlen++;
t[2 & HIBYTE(nlen)] = c;
nlen++;
t[3 & HIBYTE(nlen)] = c;
....
```

(at some point, nlen will become > 0 and thus HIBYTE goes from 0xff to 0x00, thus "grounding" indices).

Also, I would like to try to explain why I am trying to beat Fill3T. It is about those switches, while

```
switch(len) {
case 0:
case 1:
case 2:
default:
}
```

looks magnificent, it is actually 2 "unstable" branch predictions and quite a bit of code to compute the target address. So

```
if(len & 2) {
}
if(len & 1) {
}
```

should be on par - 2 branch predictions and maybe a bit less of code....

Mirek