
Subject: Re: Zooming layouts and different behaviour windows/linux

Posted by [Sender Ghost](#) on Sat, 27 Aug 2011 08:53:14 GMT

[View Forum Message](#) <> [Reply to Message](#)

I have tried your code with RTIMING on Windows XP with different compilers (TDM GCC v4.5.2 and C/C++ compiler from Microsoft Visual Studio 2008 (MSC9)) for Optimal build mode, with following results (for 5 invocations, after and before changes):

TDM GCC v4.5.2:

after changes:

```
TIMING chsync      : 4.54 s - 4.54 s ( 4.54 s / 1 ), min: 4.54 s , max: 4.54 s , nesting: 1 - 1
TIMING chsync      : 4.53 s - 4.53 s ( 4.53 s / 1 ), min: 4.53 s , max: 4.53 s , nesting: 1 - 1
TIMING chsync      : 4.52 s - 4.52 s ( 4.53 s / 1 ), min: 4.53 s , max: 4.53 s , nesting: 1 - 1
TIMING chsync      : 4.55 s - 4.55 s ( 4.55 s / 1 ), min: 4.55 s , max: 4.55 s , nesting: 1 - 1
TIMING chsync      : 4.55 s - 4.55 s ( 4.55 s / 1 ), min: 4.55 s , max: 4.55 s , nesting: 1 - 1
```

before changes:

```
TIMING chsync      : 4.45 s - 4.45 s ( 4.45 s / 1 ), min: 4.45 s , max: 4.45 s , nesting: 1 - 1
TIMING chsync      : 4.45 s - 4.45 s ( 4.45 s / 1 ), min: 4.45 s , max: 4.45 s , nesting: 1 - 1
TIMING chsync      : 4.46 s - 4.46 s ( 4.46 s / 1 ), min: 4.46 s , max: 4.46 s , nesting: 1 - 1
TIMING chsync      : 4.45 s - 4.45 s ( 4.45 s / 1 ), min: 4.45 s , max: 4.45 s , nesting: 1 - 1
TIMING chsync      : 4.44 s - 4.44 s ( 4.45 s / 1 ), min: 4.45 s , max: 4.45 s , nesting: 1 - 1
```

MSC9:

after changes:

```
TIMING chsync      : 2.98 s - 2.98 s ( 2.98 s / 1 ), min: 2.98 s , max: 2.98 s , nesting: 1 - 1
TIMING chsync      : 2.99 s - 2.99 s ( 2.99 s / 1 ), min: 2.99 s , max: 2.99 s , nesting: 1 - 1
TIMING chsync      : 2.99 s - 2.99 s ( 2.99 s / 1 ), min: 2.99 s , max: 2.99 s , nesting: 1 - 1
TIMING chsync      : 2.99 s - 2.99 s ( 2.99 s / 1 ), min: 2.99 s , max: 2.99 s , nesting: 1 - 1
TIMING chsync      : 2.99 s - 2.99 s ( 2.99 s / 1 ), min: 2.99 s , max: 2.99 s , nesting: 1 - 1
```

before changes:

```
TIMING chsync      : 2.92 s - 2.92 s ( 2.92 s / 1 ), min: 2.92 s , max: 2.92 s , nesting: 1 - 1
TIMING chsync      : 2.94 s - 2.94 s ( 2.94 s / 1 ), min: 2.94 s , max: 2.94 s , nesting: 1 - 1
TIMING chsync      : 2.93 s - 2.93 s ( 2.94 s / 1 ), min: 2.94 s , max: 2.94 s , nesting: 1 - 1
TIMING chsync      : 2.95 s - 2.95 s ( 2.95 s / 1 ), min: 2.95 s , max: 2.95 s , nesting: 1 - 1
TIMING chsync      : 2.93 s - 2.93 s ( 2.93 s / 1 ), min: 2.93 s , max: 2.93 s , nesting: 1 - 1
```

As I already said, the differences are minor (about 2-3%) in this case. I just suggested to apply font changes "at once", when needed, instead of current approach (in constructor, for each Ctrl). This is "semi automatic" method with its (dis)advantages (compared to current "automatic" method), but implementation could be different.

In conclusion, because current approach exists and differences of changes are minor, there is no need "to fix the fix". On the other hand, it was informative.
