Subject: Re: Zooming layouts and different behaviour windows/linux
Posted by Sender Ghost on Thu, 18 Aug 2011 06:24:27 GMT

View Forum Message <> Reply to Message

mdelfede wrote on Thu, 18 August 2011 01:34
Well, I've not seen how much overhead is added, but I guess that a check on a boolean before control creation (which I guess it's all needed to do the job...) is no noticeable overhead.

Well, may be on Linux it is not very noticable (Ctrl constructor -> SyncCh -> GetStdFont -> InitStdFont).

At first it just felt differently, after starting TheIDE, compiled with new changes. Then I created some (unrealistic) testing application:

```
#include <CtrlLib/CtrlLib.h>

using namespace Upp;

GUI_APP_MAIN
{
 Array<EditString> list;
 TimeStop e;

 const int n = 1000000;
 for (int i=0; i < n; ++i)
  list.Add();

 PromptOK(AsString(e.Elapsed()));
}
```

The results for changed source code:
4406
For not changed:
4266

Agree, not much, just 3% difference.

mdelfede wrote on Thu, 18 August 2011 01:34
Your solution, with INITBLOCK, could work too.... but how do you know that INITBLOCK comes *before* chamaleon init ? And, worse,
how do you know your INITBLOCK comes *after* gui init, X11 in particular, without which we could have a crash, I guess ?

I already had bad experiences with INITBLOCKS expecting something already initialized which it wasn't.....

As about INITBLOCK:
"Upp provides a nice solution to do init / deinit work of your package's static or global stuff (i.e. if

it's not Plain Ol' Data and needs some function calls).
If your INITBLOCK / EXITBLOCK resides in a .cpp file and the file contains code that is actually referenced (used) somewhere else, everything works as expected, no precautions need to be taken. If not, the linker will drop the file, your init code won't be invoked.
This is because the INITBLOCK / EXITBLOCK actually registers itself in an init facility from upper Upp code. So no code ref downwards is added."

Briefly, it depends from building order, where other INITBLOCKs (residing inside *.icpp files) linked before main code.

mdelfede wrote on Thu, 18 August 2011 01:34
What I've seen is that my app got too different layout sizes between linux and windows, and I needed a more uniform aspect;
that's more important than keeping default gui font size in this case, so I needed a way to change it at startup.

I think, this is because different DPI used for Linux and Windows environments by default. For example, I already increased DPI value on Windows for TFT monitor, but it feels much bigger for CRT monitor. Also the font sizes by default: 8 for Windows, 10 for Linux environment, such as GNOME.