
Subject: Re: Styles and Widgets
Posted by [mirek](#) on Wed, 14 Mar 2007 13:54:27 GMT
[View Forum Message](#) <> [Reply to Message](#)

WebChaot wrote on Wed, 14 March 2007 07:15Hi all!

I made some new components and changes in a previous version (610 dev 1). This week my colleague downloaded the last release (2007) and I saw, that some changes are done on chameleon technology.

1. I have to paint some widgets by code. Is there any example, how to do this in the new way (Style().Write() ??)?

It is not really too much different.

There were two reason to changes things:

"index based" method of working with style became to be overly complicated, therefore we changed indicies for structures. Usually, those structures are pretty easily understandable.

If you had any code based on indicies, conversion to structures should be pretty straightforward, usually there is one structure data fiels per former index...

While being at it, structures also provide a good way how to allow per-thread customization. So you can either assign a reference to the style for individual widget, or you can use "Write" to change the style for the whole class.

Quote:

2. I didn't look in deep of new release. But what I missed in older version is a style for StatusBar and something for "CoolBar" (the docking panel where windows toolbars are placed). I know, there is still no CoolBar and no Docking ability in upp for now. But maybe the style could be included and for example be assigned on a dummy StaticRect or something else.

OK.

Quote:

3. I put also my third question here, but its not really a chameleon topic: I want create widgets at runtime. Creation is not the problem (I think) - but how to iterate over all widgets of a window for example?

Ctrl::GetFirstChild, Ctrl::GetNext

Quote:

How to detect its class?

RTTI if you really need to.

Quote:

The reason is simple: I have to collect input data of all edit fields - but because they are created by code I need a solution to iterate over all fields on a window.

Ah. Well, there are better solution to this. See DynamicDlg example. In nutshell, you can create widgets in container, e.g. ArrayMap. GUI never owns widgets, therefore you can still use them in the dialog, but use container to access individual editors.

Mirek
