Subject: Re: First Impressions
Posted by mirek on Tue, 28 Feb 2006 09:45:19 GMT
View Forum Message <> Reply to Message

Hi,

thanks for hints, they will be useful when creating documentation...

Now some explanations:

* all buttons in LayoutDesigner which are active are indeed active Not sure what you are refering by "Alignment" buttons, but they work. By "Alignment button" I can in fact imagine two different sets: One aligns group of buttons according to some other button (say center etc). Other group defines how layout should behave when its size is changeing (to what layout edge is Ctrl edge bound).

* Layout desinger contains just the most often used Ctrl classes (this can be extended by providing .usc files) however, as you have learned the hard way, allows you add item with any class-name you wish, withou actually testing whether such class even exists or is Ctrl-derived (read widget) class. This is extremly helpful as you can easily put your own widget class into layout without problems, but here obviously resulted in newbie confusion, as Image is NOT Ctrl derived class and cannot be put to the layout. To have raster picture in Layout, use e.g. ImageCtrl.

* <temp-aux> is a "pseudo-package". The problem is that files in "regular" packages (those in main package and in transitive closure of "used" packages) get compiled into the result. OTOH, you need a way how to edit/view files that are not part of your project. Therefore those <*-aux> packages. <temp-aux> contains "temporary" files (temp-aux is empty at each TheIDE start). <prj-aux> retains content for given main package. <ide-aux> retains content across main packages.

* To add files, just click on package you want to add it (most likely your main package), then right-click files list and choose "add package directory file".

"SetInfo" - please ignore for now. That was intended to provide description of widget, however never actually used - will probably be removed soon. (And yes, it is wordprocessor).

ctx, id - those are language translations related. Read explanation of i18n.

"Generate code" - code is put into the clipboard (this is currently the rule of most code generation in U++, might improve in some cases in future, but not likely in this case)

Quote:
As usual please don't take these comments as harsh critisum, It is meant to be contructive suggestions, while I'm still a newbie.

Sure, you are welcome. U++ is not very well documented and IS radically different from anything

you have used so far - but most of those differences are there to support modularity of the codebase.

OTOH, I suggest you to read some topics in TheIDE help - I believe that some question raised are in fact described there quite well...

Mirek
