Subject: Re: Compound New <:PACKAGE:> name etc. [FEATURE REQUEST]
Posted by fudadmin on Thu, 18 May 2006 19:11:07 GMT
View Forum Message <> Reply to Message

forlano wrote on Thu, 18 May 2006 11:27If I've understood the structure of the code you have generated for my app was done on the fly with your template. It is very useful. In fact I've not touched it in that it was well organized. My opinion is favorable.

Luigi

PS: Each time I add a new *.cpp file the important include appear automatically. Even that is due to your template? What I've to modify if I want to add automatically a new include when adding a new cpp file?

Yes, I'm very glad you have asked this particular question... . Some of the problems I'm trying to solve is that with my system it would be possible:
1. - with Esc and/or current macro system analyzing the existing template and/or the package/s generated - to add new files and some/all changes to a new version template-package and automatically sort out all the dependancies.
I hate when I have to edit all those #include by hand. Especially when I want to addopt new packages to U++. I want a kind of inteligent "
template-package-template-version-incremental-upgrader-inclu der " system. Current Packages are too big pieces for me (Java and interpreters clearly have their advantages over C++...). And what if I need a several combinations of Core or etc.? Or to maintain a lot of very similar versions but individual versions to different clients? And because Mirek is "not interested" in many changes I have to maintain my own theide and libs versions. This is natural. One size can't fit all. Maybe not ideal extreme would be to have all the code templated in a database and to assembly (also visually!...) like Lego(tm)... .
2. - even more power would come if connected with uvs/svn.
3. - instead of sending a package or svn updates the users would send some params to click to generate a package...
4. - instead like a new user Luigi spending (1 month?) to create a skeleton for his program consisting of "all-must-have" nearly standard menus, tabs , status bar, file opening-saving, help system, etc. etc. he could have it all with one click... or adjust it with several clicks and param entries.
And immediately (after ~1 min!..) start playing with his data, controls and callbacks and customizing it...
5. - from "create a new package" it would work like a kind of
"Programmable-Program-Structure-Layout-Designer".
You can call it a "fancy generator" but it would save quite a lot of time even for professional users, especially, generating menus. Meanwhile, for the new users it could serve like an "incremental-tutorial-in-action+code-snippets-database".
6. - from inside theide I want it to work like a Templator++... by inserting/including name-parameter-parametrized metods/pieces of code.
7. - similarly to WideStudio it would be able to produce code for different programming languages, GUI toolkits and/or translate from each other. But to achieve that fast maybe I need something more productive than "the most productive language in history and its tools..."