Subject: Feature Request TheIDE: Refactoring / crossreference / include nesting visualization graph
Posted by jonzun on Wed, 15 Feb 2012 02:43:22 GMT
View Forum Message <> Reply to Message

Hello everybody!

I'm new here, and know about TheIDE just 2 days, so forgive me if i request something whis is already solved or rejected.

I am very impressed of the Ultimate++ / TheIDE thing.
There are so many fabulous things realized in it, i was always dreaming about. Its compact and clean. A piece of art

In the last 5 years I worked primarily with Visual Studio 2005 and the last year a bit with Eclipse CDT. In Visual Studio I learned to love the Visual Assist X plugin (http://www.wholetomato.com)

But what I always missed in most of the other IDEs editors (and so even TheIDE) are the following things:

---
1. Source code refactoring support:

Example: I rename a class member by selecting it with the mouse and select rename in a popupmenu. The IDE then replaces in all files of the project just those symbols which are real member references inside the source. Lets say all the methodes of the class accessing that member, but not similar named symbols of some other class. I mean really namespace/scope aware.

Visual Assist X (and i think CDT too) then gives you a dialog with a tree list (like your OptionTree control) showing you a preview of what its going to do to all the files. There you can manually correct and modify what will be renamed to your personal likings.

---
2. Symbol crossreference:

What everybody has is goto definition/declaration. But what really is helpful while analyzing some complex / messy code is to find all references of a SYMBOL (methode, define macro etc , of course namespace/scope aware). You can traverse the source in a tree like fashion, like you would surf the net with a web broswer. Back and forth.
Searching for all referencing locations gives a result list in the console area. There you can pick where to jump to. At the destination you can repeat this with another symbol. Then Another result list is generated and stacked on the previous one. Pressing a back button moves you back to the previous place and also to the previous result list.

CDT and Visual Assist X offer this to some degree and i need it a lot.

---
3. #include file nesting visualization graph:

Sometimes (usually always) you get not-so-nice source code to change or maintain. Usually this code as ridiculous #include nestings you simply cannot cleanup in a reasonable time without a graphical view of dependencies. The only 2 tools i knew and liked for that was Sniff+ and Sourcenav. Both out of fashion and out of reach for me.
I miss this a lot.

So i want to ask you, what do you think about adding this features to TheIDE ?

PS: If you don't throw rocks at me after this, ill come up with even more stuff

kind regards

jz

Subject: Re: Feature Request TheIDE: Refactoring / crossreference / include nesting visualization graph
Posted by dolik.rce on Wed, 15 Feb 2012 06:19:26 GMT
View Forum Message <> Reply to Message

Welcome to the forum, Jonzun

1.) Source refactoring support is not present now, but that might change. Few people are trying to incorporate LLVM based code parser into theide which would allow such things. You are second person to request this feature in quite short time, I guess it is quite popular

2.) Theide supports crossreferences. The commands are hidden in Assist menu, the default keyboard shortcuts are Alt+J and Alt+G. There are also other interesting functions useful for traversing large codebase, like history browsing: Alt+Left/Right arrow will take you back/forth to the places in code you were editing before.

3.) Analyzing the include graph would pretty easy to implement. The hard part is the display, drawing nice graphs is not trivial I usually use doxygen for this, it can generate a lot of interesting informations and schemas (requires graphviz to generate the diagrams).

Best regards,
Honza

Subject: Re: Feature Request TheIDE: Refactoring / crossreference / include nesting visualization graph
Posted by jonzun on Fri, 17 Feb 2012 13:55:08 GMT
View Forum Message <> Reply to Message

Hi!

SNIFF+ does #include graphs like here http://www-sst.informatik.tu-cottbus.de/~ssttool/sniff.htm and for detail here http://www-sst.informatik.tu-cottbus.de/~ssttool/Documentati on/ref-inc1.htm#750690

It would be helpful to visualize the "weight of usage" of each header file in the project or the whole workspace:
* The include file name in the graphical node has a bolder and/or bigger font type the more often the header file was parsed during a full rebuild.
* Find the most frequented sections of the include graph across all dependent modules.

Sure there may be manual workarounds and with some research you can fiddle your ways to the answers as well. But it takes time, is mostly unreliable and distracts the developer from the actual goal - implement software.

Whats is more problematic are those #ifdef conditional source code sections. You don't have one, but a set of possible source codes to cope with. Visual Studio does this in 2005 nice for syntax coloring. For example - You change a #define and affected code passages are greyed out (Wonderful). Greatly enhances the reading of the source code.

Crossreference: As an "Eclipse/Visual Assist X user" i was searching for crossreference features in the RMB popup menu . I definitely need more time to learn TheIDE before i can comment on this.

See SNIFF+ crossreference for an inspiration: http://www-sst.informatik.tu-cottbus.de/~ssttool/Documentati on/ref-cro1.htm#655139 . I don't liked all of it. But some ideas are worth thinking about.

Generally "analysing source" workflow in SNIFF+ had the popup panel/windows overkill problem. After some time your destop was filled with all kinds of views. The docked panels of todays IDEs do a better job there. But on the other side this way you "naturally" build a stack of panels which you can traverse back and forth by opening and closing them.

(in TheIDE you could optionally use the editor split view to display the graph in one side.)

Yes I realize now, I miss SNIFF+

The final, generally true statement: Damn if I had more time i would do this by myself

kind regards

jz