Subject: debugger wishlist
Posted by iplayfast on Fri, 24 Feb 2006 07:39:15 GMT
View Forum Message <> Reply to Message

1. Breakpoint selection in the gutter. Click the gutter and a dot appears. Click again it goes away. You can get cute and add all sorts of functionality to it, but the basic breakpoint is really missed.

2. Mouse hover shows variables.

Subject: Re: debugger wishlist
Posted by mirek on Fri, 24 Feb 2006 09:59:00 GMT
View Forum Message <> Reply to Message

What is the "gutter" ?

BTW, my wishlist for debugger/linux is makeing it "native" - bypassing GDB and doing stuff directly. That is how MSC debugger works now, using dbghelp.dll API. The missing part is equivalent library (and needed knowhow) for GDB.

Mirek

Subject: Re: debugger wishlist
Posted by iplayfast on Fri, 24 Feb 2006 16:47:37 GMT
View Forum Message <> Reply to Message

The gutter is the white area between the left side of the code and the frame. In this ide it's very small (about 1/8" on my monitor) on Borlands it is large enough to show line numbers.

I'm not sure what the advantage of going native is. If the gbd is running in the background but you can't actually tell, then what's the difference?

Plus, are you planning on supporting multiple processors. You will be spending all your time writting the debugger. I think the strategy you are currently using is pretty good.

Subject: Re: debugger wishlist
Posted by mirek on Fri, 24 Feb 2006 17:01:01 GMT
View Forum Message <> Reply to Message

iplayfast wrote on Fri, 24 February 2006 11:47The gutter is the white area between the left side of the code and the frame. In this ide it's very small (about 1/8" on my monitor) on Borlands it is large enough to show line numbers.

I'm not sure what the advantage of going native is. If the gbd is running in the background but you can't actually tell, then what's the difference?

Plus, are you planning on supporting multiple processors. You will be spending all your time writting the debugger. I think the strategy you are currently using is pretty good.

Well, in that case you, breakpoints should behave exactly as you wish (doubleclick the gutter).

You can even have line numbers, just activate them in environment settings (personally I prefer to have more space for source text, but...)

The GDB problem is that text communication is very slow with it. E.g. when you are about to display any widget class (using "print" command in GDB), GDB feeds you with more than 100KB data that you have to parse. Now if you have more such variables to display, things can get awfuly slow.

If only GDB supported giving you type information, I would rather used raw hex dumps of memory.... however, U++ debugger for MSC compiler is near to ideal solution.

Mirek

Subject: Re: debugger wishlist
Posted by iplayfast on Fri, 24 Feb 2006 17:24:50 GMT
View Forum Message <> Reply to Message

luzr wrote on Fri, 24 February 2006 12:01
Well, in that case you, breakpoints should behave exactly as you wish (doubleclick the gutter).

You can even have line numbers, just activate them in environment settings (personally I prefer to have more space for source text, but...)

Mirek

Missed that with the double clicking. You guys are fast at granting wishes I don't normally want line numbers, I was just using that as an example. However now that you mention it, it would be nice to be able to toggle that on/off in the context menu.

Wishlist of context menu when editing
Help on topic under cursor (should be mapped to same as F1)
goto line
Line numbers checkbox
When the cursor is in a function/class/structure name Go to declaration
Go to Implementation
Bookmark with submenu allowing you to check/uncheck up to 10 different bookmarks
Goto Bookmark with submenu (matching above).

Add Todo

Code Warrior has a nice feature that becomes addictive. A dropdown list of functions in the file that allows you to go to the top of the function you select.
In the .h it would be useful to have the same only for classes and structures.

(Aren't you sorry you granted my breakpoint wish

Oh and I also wish that the breakpoint toggle was a single click.

Subject: Re: debugger wishlist
Posted by mirek on Fri, 24 Feb 2006 17:35:19 GMT
View Forum Message <> Reply to Message

iplayfast wrote on Fri, 24 February 2006 12:24luzr wrote on Fri, 24 February 2006 12:01
Well, in that case you, breakpoints should behave exactly as you wish (doubleclick the gutter).

You can even have line numbers, just activate them in environment settings (personally I prefer to have more space for source text, but...)

Mirek

Missed that with the double clicking. You guys are fast at granting wishes I don't normally want line numbers, I was just using that as an example. However now that you mention it, it would be nice to be able to toggle that on/off in the context menu.

OK, puttin' to ToDo...

However, we absolutely need to concetrate on serious bugs now to be able to launch 602 (511 is too out-dated now).

Mirek

Subject: Re: debugger wishlist
Posted by iplayfast on Fri, 24 Feb 2006 17:39:04 GMT
View Forum Message <> Reply to Message

You respond so fast. I edited the original wish and added some more. I understand that these are wishes and will be fit in (or not) as the schedule allows.

Subject: Re: debugger wishlist
Posted by mirek on Fri, 24 Feb 2006 17:44:53 GMT

I am not sure what version are you using, but 602:

Quote:
When the cursor is in a function/class/structure name Go to declaration

Alt+I

Quote:
Go to Implementation

Alt+J

Quote:
Bookmark with submenu allowing you to check/uncheck up to 10 different bookmarks

File/Bookmarks, altso Ctrl+Shift+0-9 to set, without Shift to go.

Quote:
Code Warrior has a nice feature that becomes addictive. A dropdown list of functions in the file that allows you to go to the top of the function you select.

Ctrl+G. Use for goto line as well.

Quote:
Oh and I also wish that the breakpoint toggle was a single click.

OK Will add an option. (Personally, I prefer F9).

Mirek

Subject: Re: debugger wishlist
Posted by iplayfast on Fri, 24 Feb 2006 18:33:22 GMT

That's all very useful info. I wouldn't have guessed at all that functionality being available. Which is my point.
The functionallity should be made more visible by putting it into the context menu. (Ie right click in the source code and see all this stuff you've got).

The fact that there is key assignments for it should also be shown in the context menu. Then people can learn to use your software with menus, but as they get more used to it, they can switch to keys combinations.

Subject: Re: debugger wishlist
Posted by mirek on Fri, 24 Feb 2006 20:51:22 GMT
View Forum Message <> Reply to Message

iplayfast wrote on Fri, 24 February 2006 13:33That's all very useful info. I wouldn't have guessed at all that functionality being available. Which is my point.
The functionallity should be made more visible by putting it into the context menu. (Ie right click in the source code and see all this stuff you've got).

The fact that there is key assignments for it should also be shown in the context menu. Then people can learn to use your software with menus, but as they get more used to it, they can switch to keys combinations.

Well, it is all (or most of it) in Assist menu, but you have a point here, context menu in source editor could be much more developed. In ToDo

Mirek

Subject: Re: debugger wishlist
Posted by wilho on Mon, 27 Feb 2006 22:24:01 GMT
View Forum Message <> Reply to Message

call stack viewer would be nice...

Subject: Re: debugger wishlist
Posted by mirek on Mon, 27 Feb 2006 22:35:04 GMT
View Forum Message <> Reply to Message

wilho wrote on Mon, 27 February 2006 17:24call stack viewer would be nice...

It expands from that droplist with function name.

Mirek

Subject: Re: debugger wishlist
Posted by wilho on Mon, 27 Feb 2006 22:47:18 GMT

And I wondered how in earth there isn't one

Thanks.

Subject: Re: debugger wishlist
Posted by mr_ped on Tue, 07 Mar 2006 16:08:28 GMT

I can think of many improvements to debugger, but many of them are not important at all for Ultimate++.
Yet in case you will have some mood to improve debugger, here are some ideas.
(I was testing with 602 .. BTW, I'm unable to get version number inside of TheIDE, there's no "about" box in menu)

- a way to format values per variable in Watches.
(i.e. "CheckSum,h" in Watches would produce value of CheckSum variable, but shown as hexa number)
(it's not only about hexa, in MS VS it was sometimes crucial to specify number of array elements to show, etc...)

BTW, I prefer hexa values as default, some global switch in debug would be nice too.

- in ASM view there are so many things missing...
 flags, FPU regs
 hints about final value (i.e. [ebp-0x4] should have somewhere hint "0xaddress : 0xvalue", where address is ebp-4, and value the memory content), etc... (just like "Autos" do work with C source)
 Editing values of system registers (can't be done trough Watches too, as they don't know "EAX")
 I know this is not ASM debugger, so these are not important at all, yet if your ToDo will be too empty, there's lot of room for improvements here.

- improved Autos, if you have in source for example:

```
void foo(void) {
  long tab[10], i, j;
  /* ... */
  for (i=0; i < 9; ++i)
    j = tab[i] + tab[i+1];
}
```

Let the Autos show also value inside of array (tab[i] and tab[i+1]), not just "tab" address and i value.
 actually "tab" is shown as "address->first_value", there's no sign of array.
 "tab[3]" in Watches does produce "Only pointer can be dereferenced" (this is connected to formated output too)

- improved Memory view:
 Goto - if I enter "i", it will go to "&i", not to value of "i".
 Ability to switch between 8bit/16bit/32b/64b/128b/256b values.
 It's sort of pity to read long 0x12345678 like 0x78 0x56 0x34 0x12 (on Intel).
 Multiple "memory view" windows (undockable), (this one would be nice also with "watches", especially if you are comparing for example two different instances of the same class, it's nice to have them as tiled windows)

- conditional breakpoints
 (it was always a pain for me to make them work in MS VS, but in the end they helped a lot some times)

WAIT A SEC!
I can *not* change value of anything in Autos/Locals/Watches.
Now this one is major!
If I start to debug some code, and I notice some error which does produce bad input to some other function, I often like to correct it by hand in debugger, and continue further to check for more errors. Having to fix the problem, recompile, restart, go to the desired place is IMHO too much hassle.
 Actually having the ability to change values is nice also for testing extreme conditions without modifying sources to call your functions under such condition.

I can not modify memory also!

Well, enough for today.

Subject: Re: debugger wishlist
Posted by mirek on Tue, 07 Mar 2006 21:12:56 GMT
View Forum Message <> Reply to Message

mr_ped wrote on Tue, 07 March 2006 11:08I can think of many improvements to debugger, but many of them are not important at all for Ultimate++.
Yet in case you will have some mood to improve debugger, here are some ideas.

Just for the record, are you speaking about GCC/GDB or MSC?

Mirek

Subject: Re: debugger wishlist
Posted by mr_ped on Wed, 08 Mar 2006 13:53:18 GMT
View Forum Message <> Reply to Message

UPS...

WinXP + MS VC++ Toolkit 2003

Subject: Re: debugger wishlist
Posted by hojtsy on Fri, 10 Mar 2006 22:21:18 GMT
View Forum Message <> Reply to Message

Sometimes the value of a watched variable is very long and gets truncated by the window edge. I would like that to be broken up into multiple lines. There are also some variables which seem to be truncated in the middle of the line (bug?), I would like to see the full value.
And displaying in tooltip the value of variables hovered in the editor would be quite cool.

Subject: Re: debugger wishlist
Posted by unodgs on Sat, 11 Mar 2006 10:01:23 GMT
View Forum Message <> Reply to Message

hojtsy wrote on Fri, 10 March 2006 17:21Sometimes the value of a watched variable is very long and gets truncated by the window edge. I would like that to be broken up into multiple lines. There are also some variables which seem to be truncated in the middle of the line (bug?), I would like to see the full value.
And displaying in tooltip the value of variables hovered in the editor would be quite cool.

I second all that!
