
Subject: Re: debugger wishlist

Posted by [mr_ped](#) on Tue, 07 Mar 2006 16:08:28 GMT

[View Forum Message](#) <> [Reply to Message](#)

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 formatted 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.
