Subject: the debugger
Posted by cocob on Fri, 08 Aug 2008 18:55:45 GMT
View Forum Message <> Reply to Message

Hello all,

Excuse me for my impatience ! But i want to know if improvement in theide debugger are envisaged.

I know this is not the priority but i'm using theide for some projects since 6 months and the debugger is for me the only thing which need to be improved.

I have always same problem.

- Add/remove breakpoints during execution doesn't work.
- Sometimes my program or/and theide is stucked.
- watching values is limited and it would be nice to show
  expression like b.size() or b[index] (where b is a STL container)
  (if it is not possible it would be really nice the debugging session was not broken adding expressions like these.)
- and some others little problems, ...

thanks for your answer.

cocob

Subject: Re: the debugger
Posted by zsolt on Fri, 08 Aug 2008 19:40:03 GMT
View Forum Message <> Reply to Message

Quote:
- Add/remove breakpoints during execution doesn't work.
Try F9 key.

My favorite feature would be the ability to show content of Value types.

Subject: Re: the debugger
Posted by cocob on Fri, 08 Aug 2008 20:21:06 GMT
View Forum Message <> Reply to Message

ok but F9 show/add a red point at the left of my code but the programs doesn't stop on it or stop on it after it is removed ....

Subject: Re: the debugger
Posted by mirek on Fri, 08 Aug 2008 22:52:43 GMT
View Forum Message <> Reply to Message

You have forgot to mention on what platform.

There are two things. With MSC, we are sort of victims of information that dbghelp.dll reveals us about the executable.

With GCC, we are victims of GDB....

Mirek

Subject: Re: the debugger
Posted by cocob on Fri, 15 Aug 2008 10:32:56 GMT
View Forum Message <> Reply to Message

I wants to know why gdb is a problem.
I'am not expert in debuggers but with gdb you can display values of very complex types adding some scripts or complex expression like these

display *(vec._M_impl._M_start + 3)
-> equivalent to vec[3]

or

define vector
printf "size=%d\n", $arg0._M_impl._M_finish - $arg0._M_impl._M_start
printf "capacity=%d\n", $arg0._M_impl._M_end_of_storage - $arg0._M_impl._M_start
end

to display size and capacity of a vector

This is examples but for me a lot a of expression can be evaluate by gdb and it would be great to do the same in theide !

cocob

Subject: Re: the debugger
Posted by cocob on Sat, 11 Oct 2008 09:26:23 GMT
View Forum Message <> Reply to Message

I have done some test with theIDE on windows with MSC9

Here my example source :

```
#include <vector>

int main(int argc, const char *argv[])
{
 std::vector<int> vec;
 std::vector<int>::iterator iter;

 vec.push_back(1);
 vec.push_back(4);
 vec.push_back(3);

 iter = vec.begin();
 int a = *iter;
 return 0;
}
```

I have tried to watch these expressions

1 - (vec._Mylast - vec._Myfirst)
2 - vec.size()
3 - vec[0]
4 - *iter
5 - *(vec._Myfirst+1)

and the results

1 - ff698284->??
2 - Unknown membre 'size'
3 - Invalid operand
4 - Only pointer can be dereferenced
5 - 4

So only the last expression works.
I know that 2, 3, 4 are difficult because we sould parse c++, detect specific STL type (vector) and replace the expression.

But why the first expression doesn't work ?

cocob

Subject: Re: the debugger
Posted by cocob on Sat, 18 Oct 2008 10:40:45 GMT
View Forum Message <> Reply to Message

Here a little code to solve my problem of printing values in STL vectors. I think it would be great to improve this to print values for other types.

```
String Gdb::GetVarType(const String& var)
{
 String type = FastCmd("ptype " + var);
 StringStream ss(type);
 String ln = ss.GetLine();
 const char *s = strchr(ln, '=');
 if(s) {
  s++;
  while(*s == ' ')
   s++;
  return String(s);
 }
 else
  return "";
}

String Gdb::TransformExp(const String& exp)
{
 String newExp;
 int b = exp.Find('[');
 if(b>0) {
  int e = exp.Find(']', b+1);
  if(b>0) {
   // we have a *[*] expression so evaluate the index
   String index = exp.Mid(b+1, e-b-1);
   index = Print(index);
   String tab = exp.Mid(0, b);
   String tabType = GetVarType(tab);
   if(tabType.GetLength()==0) return exp;
   if(tabType.Left(17)=="class std::vector") {
    newExp << "*(" << tab << "._M_impl._M_start+" << index << ")";
    return newExp;
   } else {
    newExp << tab << "[" << index << "]";
    return newExp;
   }
  }
 }

 return exp;
}

String Gdb::Print(const String& exp)
{
 String newExp = TransformExp(exp);
 String q = FastCmd("print " + newExp);
 StringStream ss(q);
```

```
 String ln = ss.GetLine();
 const char *s = strchr(ln, '=');
 if(s) {
  s++;
  while(*s == ' ')
   s++;
  return DataClean(s);
 }
 else
  return DataClean(ln);
}
```

Subject: Re: the debugger
Posted by cocob on Wed, 07 Jan 2009 15:45:11 GMT
View Forum Message <> Reply to Message

I have an other problem but very simple.

int n[4];

Impossible to display array values with MSC debugger...

Subject: Re: the debugger
Posted by Mindtraveller on Mon, 27 Apr 2009 17:29:20 GMT
View Forum Message <> Reply to Message

Are there any plans to support setting/clearing breakpoints during the debug session (@ at least one of MSC/GDB)? This will be very handy.

Subject: Re: the debugger
Posted by Mindtraveller on Sun, 03 May 2009 15:28:58 GMT
View Forum Message <> Reply to Message

It looks like adding/removing breakpoints while in debug mode works in some cases. But in some cases - it doesn`t.
Looks like a bug(?).

Subject: Re: the debugger
Posted by sergeynikitin on Sun, 03 May 2009 19:10:38 GMT
View Forum Message <> Reply to Message

Which operating system and what compiler is used?

I use mostly GCC in UBUNTU-i386 and the GCC / MINGW in Windows XP. Nothing like I have not noticed. Apparently the case in your environment.

Subject: Re: the debugger
Posted by Mindtraveller on Sun, 03 May 2009 20:29:01 GMT
View Forum Message <> Reply to Message

WinXP SP2,
MSC8, MSC9
