Subject: Problems debugging with Visual C++
Posted by frankdeprins on Thu, 11 Sep 2014 06:32:47 GMT
View Forum Message <> Reply to Message

Hello Mirek,

After a very long time, I just started to do some c++ again and, hence, use theIDE. But I had quite some problems debugging with Visual C++ (2010/32 bit).
Amongst the problems are crashing (I just upgraded to rev 7655 yesterday, but did not test that one a lot) and inability to watch variables (also with 7655). All my variables are either 0 (numeric ones) or marked with '??'.
Are there any added preconditions for debugging with VC now?

Best regards,

Frank

Subject: Re: Problems debugging with Visual C++
Posted by mirek on Tue, 16 Sep 2014 12:12:27 GMT
View Forum Message <> Reply to Message

frankdeprins wrote on Thu, 11 September 2014 08:32Hello Mirek,

After a very long time, I just started to do some c++ again and, hence, use theIDE. But I had quite some problems debugging with Visual C++ (2010/32 bit).
Amongst the problems are crashing (I just upgraded to rev 7655 yesterday, but did not test that one a lot) and inability to watch variables (also with 7655). All my variables are either 0 (numeric ones) or marked with '??'.
Are there any added preconditions for debugging with VC now?

Best regards,

Frank

Debugger was going through intense development lately. Currently, it works better than "before" for me, but it is definitely possible that something is still a bit broken. Anyway, for now I recommend testing with latest svn version (or night build).

One possible precondition could be 'current' version of dbghelp.dll. What is your OS?

It should be possible to get latest dbghelp.dll somewhere, I guess it is even in SDK, but it is possible that your OS is e.g. winxp and you have no dbghelp.dll in theide.exe dir (and thus it links with old version in system32).

Mirek

Subject: Re: Problems debugging with Visual C++
Posted by frankdeprins on Tue, 16 Sep 2014 14:02:13 GMT
View Forum Message <> Reply to Message

Hello Mirek,

I am working on Windows 7 32-bit.
I will try finding a newer dbghelp.dll and keep you informed.

Thanks,

Frank

Subject: Re: Problems debugging with Visual C++
Posted by frankdeprins on Wed, 17 Sep 2014 05:50:59 GMT
View Forum Message <> Reply to Message

Hello,

Found a newer version of the dll (6.5.xxx -> 6.12.xxx), but it made no difference.
I attached a screenshot with a simplified test scenario.

Frank

File Attachments
1) debuggertest.png, downloaded 404 times

Subject: Re: Problems debugging with Visual C++
Posted by mirek on Wed, 17 Sep 2014 13:36:02 GMT
View Forum Message <> Reply to Message

I have just spotted one issue that might cause this, can you check with latest svn please?

Other than that, I have to admit I never tested with 32-bit OS (but I did have tested theide compiled as 32 bit .exe).

Subject: Re: Problems debugging with Visual C++
Posted by mirek on Wed, 17 Sep 2014 13:41:33 GMT
View Forum Message <> Reply to Message

Just to clarify, the change is with this method:

```
BOOL CALLBACK Pdb::EnumLocals(PSYMBOL_INFO pSym, ULONG SymbolSize, PVOID
UserContext)
{
 LocalsCtx& c = *(LocalsCtx *)UserContext;

 if(pSym->Tag == SymTagFunction)
  return TRUE;

 Val& v = (pSym->Flags & IMAGEHLP_SYMBOL_INFO_PARAMETER ? c.param :
c.local).GetAdd(pSym->Name);
 v.address = (adr_t)pSym->Address;
 if(pSym->Flags & IMAGEHLP_SYMBOL_INFO_REGISTER)
  v.address = pSym->Register;
 else
 if(pSym->Flags & IMAGEHLP_SYMBOL_INFO_REGRELATIVE) {
  if(pSym->Register == CV_ALLREG_VFRAME) {
  #ifdef CPU_64
   if(c.pdb->win64)
    v.address += c.pdb->GetCpuRegister(*c.context, CV_AMD64_RBP);
   else
  #endif
    v.address += (adr_t)c.pdb->GetCpuRegister(*c.context, CV_REG_EBP);
  }
  else
   v.address += (adr_t)c.pdb->GetCpuRegister(*c.context, pSym->Register);
 }
 else
 if(pSym->Flags & IMAGEHLP_SYMBOL_INFO_FRAMERELATIVE)
  v.address += c.frame;
 c.pdb->TypeVal(v, pSym->TypeIndex, (adr_t)pSym->ModBase);
 LLOG("LOCAL " << pSym->Name << ": " << Format64Hex(v.address));
 return TRUE;
}
```

- and I believe the problems you are enduring is in there too. So perhaps if you have a bit of time and energy and my change does not work, could you please test with some RDUMPs put there, like

```
BOOL CALLBACK Pdb::EnumLocals(PSYMBOL_INFO pSym, ULONG SymbolSize, PVOID
UserContext)
{
 LocalsCtx& c = *(LocalsCtx *)UserContext;

 if(pSym->Tag == SymTagFunction)
  return TRUE;
 Val& v = (pSym->Flags & IMAGEHLP_SYMBOL_INFO_PARAMETER ? c.param :
c.local).GetAdd(pSym->Name);
 v.address = (adr_t)pSym->Address;
     RLOG("-------------------------");
      RDUMP(Format64Hex(v.address));
 if(pSym->Flags & IMAGEHLP_SYMBOL_INFO_REGISTER)
  v.address = pSym->Register;
 else
 if(pSym->Flags & IMAGEHLP_SYMBOL_INFO_REGRELATIVE) {
  if(pSym->Register == CV_ALLREG_VFRAME) {
  #ifdef CPU_64
   if(c.pdb->win64)
    v.address += c.pdb->GetCpuRegister(*c.context, CV_AMD64_RBP);
   else
  #endif
    v.address += (adr_t)c.pdb->GetCpuRegister(*c.context, CV_REG_EBP);
                    RDUMP(Format64Hex(v.address));
  }
  else
   v.address += (adr_t)c.pdb->GetCpuRegister(*c.context, pSym->Register);
 }
 else
 if(pSym->Flags & IMAGEHLP_SYMBOL_INFO_FRAMERELATIVE) {
              RDUMP(Format64Hex(c.frame));
  v.address += c.frame;
        }
 c.pdb->TypeVal(v, pSym->TypeIndex, (adr_t)pSym->ModBase);
 RLOG("LOCAL " << pSym->Name << ": " << Format64Hex(v.address));
 return TRUE;
}
```

Subject: Re: Problems debugging with Visual C++
Posted by frankdeprins on Wed, 17 Sep 2014 14:36:43 GMT

View Forum Message <> Reply to Message

Thanks Mirek,

I'm afraid it did not solve the issue, but something changed: the double variable that you can see in the screenshot, is now not listed anymore at all. Only the two integers are listed and they are still both 0.

Anyway, here are the contents of theide.log:

```
* C:\Frank\upp\theide.exe 17.09.2014 16:30:42, user: FDP
```

```
-------------------------
Format64Hex(v.address) = ffffffffffffffe8
LOCAL i64: bc
-------------------------
Format64Hex(v.address) = fffffffffffffff0
LOCAL d: c4
-------------------------
Format64Hex(v.address) = fffffffffffffffc
LOCAL i: d0
-------------------------
Format64Hex(v.address) = 8
LOCAL app: 104
-------------------------
Format64Hex(v.address) = 8
LOCAL argc: 10
-------------------------
Format64Hex(v.address) = c
LOCAL argv: 14
-------------------------
Format64Hex(v.address) = ffffffffffffffdc
LOCAL mainret: 30
-------------------------
Format64Hex(v.address) = ffffffffffffffe0
LOCAL managedapp: 34
-------------------------
Format64Hex(v.address) = ffffffffffffffe4
LOCAL initret: 38
```

Frank

Subject: Re: Problems debugging with Visual C++
Posted by mirek on Wed, 17 Sep 2014 17:53:55 GMT

View Forum Message <> Reply to Message

Well, believe or not, I think we are getting somewhere!

Normally, local variables take this path:

```
 if(pSym->Flags & IMAGEHLP_SYMBOL_INFO_REGRELATIVE) {
          if(pSym->Register == CV_ALLREG_VFRAME) {
```

(because they are relative to ebp). Obviously, it does not happen for you.

To check the theory, please

a) check both conditions: RDUMP(pSym->Flags & IMAGEHLP_SYMBOL_INFO_REGRELATIVE); RDUMP(pSym->Register ==

CV_ALLREG_VFRAME); RDUMP(pSym->Register) before first if
b) just for test, change them to 'true'

Thanks for help!

Mirek

Subject: Re: Problems debugging with Visual C++
Posted by mirek on Wed, 17 Sep 2014 17:56:40 GMT

View Forum Message <> Reply to Message

PS.: If changing to true does not help, we might also try to comment out the test for register...

```
// if(pSym->Flags & IMAGEHLP_SYMBOL_INFO_REGISTER)
//  v.address = pSym->Register;
// else
```

Mirek

Subject: Re: Problems debugging with Visual C++
Posted by frankdeprins on Thu, 18 Sep 2014 06:48:03 GMT

View Forum Message <> Reply to Message

Hello Mirek,

The extra logging lines added the next content to theide.log:

```
pSym->Flags & IMAGEHLP_SYMBOL_INFO_REGRELATIVE = 16
pSym->Register == CV_ALLREG_VFRAME = false
pSym->Register = 22
```

These log lines occurred several times, of course, but always with exactly the same values.
After that, I changed if(pSym->Register == CV_ALLREG_VFRAME) to: if(true)
But even then, the result was that I only got i and i64 and both watches were still 0.
I guess commenting out the register test will not change it, as we already know the else block is executed because of the presence of the new logging lines put in it.
Anyway, this is the final state of the code as I ran it:

```
BOOL CALLBACK Pdb::EnumLocals(PSYMBOL_INFO pSym, ULONG SymbolSize, PVOID
UserContext)
{
   LocalsCtx& c = *(LocalsCtx *)UserContext;
```

```
  if(pSym->Tag == SymTagFunction)
    return TRUE;

  Val& v = (pSym->Flags & IMAGEHLP_SYMBOL_INFO_PARAMETER ? c.param :
c.local).GetAdd(pSym->Name);
  v.address = (adr_t)pSym->Address;
RLOG("-------------------------");
RDUMP(Format64Hex(v.address));
  if(pSym->Flags & IMAGEHLP_SYMBOL_INFO_REGISTER)
    v.address = pSym->Register;
  else {
    RDUMP(pSym->Flags & IMAGEHLP_SYMBOL_INFO_REGRELATIVE);
    RDUMP(pSym->Register == CV_ALLREG_VFRAME);
    RDUMP(pSym->Register);
    if(pSym->Flags & IMAGEHLP_SYMBOL_INFO_REGRELATIVE) {
      if(true/*pSym->Register == CV_ALLREG_VFRAME*/) {
      #ifdef CPU_64
        if(c.pdb->win64)
          v.address += c.pdb->GetCpuRegister(*c.context, CV_AMD64_RBP);
        else
      #endif
          v.address += (adr_t)c.pdb->GetCpuRegister(*c.context, CV_REG_EBP);
      }
      else
        v.address += (adr_t)c.pdb->GetCpuRegister(*c.context, pSym->Register);
      RDUMP(v.address);
    }
  }
  if(pSym->Flags & IMAGEHLP_SYMBOL_INFO_FRAMERELATIVE) {
          RDUMP(Format64Hex(c.frame));
    v.address += c.frame;
     }
  c.pdb->TypeVal(v, pSym->TypeIndex, (adr_t)pSym->ModBase);
  RLOG("LOCAL " << pSym->Name << ": " << Format64Hex(v.address));
  return TRUE;
}
```

Attached, you find the log file.

Frank

File Attachments

1) theide.log, downloaded 389 times

Subject: Re: Problems debugging with Visual C++

Posted by mirek on Thu, 18 Sep 2014 08:51:35 GMT
View Forum Message <> Reply to Message

Well, that is interesting, because 22 is CV_REG_EBP, so it essentially seems to do the very same and correct thing... but it looks like the value fetched from the register is for some reason wrong.

Can we check?

```
 if(pSym->Flags & IMAGEHLP_SYMBOL_INFO_REGRELATIVE) {
  if(pSym->Register == CV_ALLREG_VFRAME) {
  #ifdef CPU_64
   if(c.pdb->win64)
    v.address += c.pdb->GetCpuRegister(*c.context, CV_AMD64_RBP);
   else
  #endif
    v.address += (adr_t)c.pdb->GetCpuRegister(*c.context, CV_REG_EBP);
  }
  else
   v.address += (adr_t)c.pdb->GetCpuRegister(*c.context, pSym->Register);
  RDUMP(Format64Hex((adr_t)c.pdb->GetCpuRegister(*c.context, pSym->Register)));
  RDUMP(Format64Hex(v.address));
 }
```

Subject: Re: Problems debugging with Visual C++
Posted by mirek on Thu, 18 Sep 2014 09:02:51 GMT
View Forum Message <> Reply to Message

BTW, is backtrace OK (except perhaps parameter values)?

Subject: Re: Problems debugging with Visual C++
Posted by frankdeprins on Thu, 18 Sep 2014 09:31:56 GMT
View Forum Message <> Reply to Message

What do you mean by backtrace?
Is it this ('Debug/Copy backtrace' menu item) what you mean:

```
ConsoleMainFn_()
Upp::AppExecute__(app=??)
main(argc=0, argv=??)
__tmainCRTStartup()
mainCRTStartup()
76faee1c (kernel32.dll)
77ac37eb (ntdll.dll)
```

77ac37be (ntdll.dll)

As a sidenote: I did pass 4 parameters via the 'Run Options' dialog, so argc should not be 0
Again, I attached the new log file.

PS: The disapearance of the double variable is only in the 'Autos' tab; in the 'Locals' tab it is still there (with '??').

File Attachments

1) theide.log, downloaded 382 times

Subject: Re: Problems debugging with Visual C++
Posted by mirek on Thu, 18 Sep 2014 10:55:30 GMT
View Forum Message <> Reply to Message

Interesting. May I ask for more logs?

```
uint64 Pdb::GetCpuRegister(const Context& ctx, int sym)
{
 int q = GetRegisterList().Find(sym);
 RLOG("== GetCpuRegister =========");
 RDUMP(sym);
 RDUMP(q);
 if(q < 0)
  return 0;
 const CpuRegister& r = GetRegisterList()[q];
 RDUMP(r.name);
 RDUMP(r.kind);
 RDUMP(r.sym);
#ifdef CPU_64
 uint64 val = win64 ? GetRegister64(ctx, sym) : GetRegister32(ctx, sym);
#else
 uint64 val = GetRegister32(ctx, sym);
#endif
 RDUMP(Format64Hex(val));
 switch(r.kind) {
 case REG_L:
  return LOBYTE(val);
 case REG_H:
  return HIBYTE(val);
 case REG_X:
  return LOWORD(val);
 case REG_E:
  return LODWORD(val);
 }
 return val;
}



BOOL CALLBACK Pdb::EnumLocals(PSYMBOL_INFO pSym, ULONG SymbolSize, PVOID
UserContext)
{
 LocalsCtx& c = *(LocalsCtx *)UserContext;

 if(pSym->Tag == SymTagFunction)
  return TRUE;

 RLOG("=== EnumLocals ========");
 RDUMP(UserContext);
 RDUMP(c.context);
 RDUMP(pSym->Register);
 Val& v = (pSym->Flags & IMAGEHLP_SYMBOL_INFO_PARAMETER ? c.param :
c.local).GetAdd(pSym->Name);
 v.address = (adr_t)pSym->Address;
 if(pSym->Flags & IMAGEHLP_SYMBOL_INFO_REGISTER)
  v.address = pSym->Register;
 else
 if(pSym->Flags & IMAGEHLP_SYMBOL_INFO_REGRELATIVE) {
  if(pSym->Register == CV_ALLREG_VFRAME) {
  #ifdef CPU_64
   if(c.pdb->win64)
    v.address += c.pdb->GetCpuRegister(*c.context, CV_AMD64_RBP);
   else
  #endif
    v.address += (adr_t)c.pdb->GetCpuRegister(*c.context, CV_REG_EBP);
  }
  else
   v.address += (adr_t)c.pdb->GetCpuRegister(*c.context, pSym->Register);
 }
 else
 if(pSym->Flags & IMAGEHLP_SYMBOL_INFO_FRAMERELATIVE)
  v.address += c.frame;
 c.pdb->TypeVal(v, pSym->TypeIndex, (adr_t)pSym->ModBase);
 LLOG("LOCAL " << pSym->Name << ": " << Format64Hex(v.address));
 return TRUE;
}
```

Thanks a lot, I feel we are really very close now.

Mirek

Subject: Re: Problems debugging with Visual C++
Posted by mirek on Thu, 18 Sep 2014 11:06:27 GMT
View Forum Message <> Reply to Message

Ops, no need.

I guess I have found it. Such a stupid bug... :(

Please try to replace this method:

```
const VectorMap<int, Pdb::CpuRegister>& Pdb::GetRegisterList()
{
 static VectorMap<int, CpuRegister> r32;
 ONCELOCK {
#define CPU_REG(sym_, context_var, kind_, name_, flags_) { CpuRegister& r = r32.Add(sym_);
r.sym = sym_; r.kind = kind_; r.name = name_; r.flags = flags_; }
  #include "i386.cpu"
#undef CPU_REG
 }
#ifdef CPU_64
 static VectorMap<int, CpuRegister> r64;
 ONCELOCK {
#define CPU_REG(sym_, context_var, kind_, name_, flags_) { CpuRegister& r = r64.Add(sym_);
r.sym = sym_; r.kind = kind_; r.name = name_; r.flags = flags_; }
  #include "amd64.cpu"
#undef CPU_REG
 }
 return win64 ? r64 : r32;
#else
 return r32;
#endif
}
```

Subject: Re: Problems debugging with Visual C++
Posted by frankdeprins on Thu, 18 Sep 2014 11:48:55 GMT
View Forum Message <> Reply to Message

Since a picture says more than a 1000 words...
Please check new screenshot.

It seems you nailed it: congratulations 8)

File Attachments

1) debuggerfixed.png, downloaded 363 times

Subject: Re: Problems debugging with Visual C++
Posted by frankdeprins on Thu, 18 Sep 2014 11:50:33 GMT
View Forum Message <> Reply to Message

And here is the latest log file

File Attachments

1) theide.log, downloaded 383 times

Subject: Re: Problems debugging with Visual C++
Posted by mirek on Thu, 18 Sep 2014 11:53:21 GMT
View Forum Message <> Reply to Message

Thanks a lot with helping me with this. It was very stochastic error (unintialized data with only 1/256 chance of bad behaviour), in theory I could have got it here too, but was not).

Subject: Re: Problems debugging with Visual C++
Posted by frankdeprins on Thu, 18 Sep 2014 12:04:07 GMT
View Forum Message <> Reply to Message

No problem at all; it's the least I can do to thank for such a great environment.

By the way: I understand you are using Upp for 64 bit windows development.
Since I got my new machine, yesterday, I wonder if the conversion from 32 bit is a smooth ride.
Are you using Visual Studio 13?

Regards,

Frank

Subject: Re: Problems debugging with Visual C++
Posted by mirek on Thu, 18 Sep 2014 21:20:57 GMT
View Forum Message <> Reply to Message

frankdeprins wrote on Thu, 18 September 2014 14:04No problem at all; it's the least I can do to thank for such a great environment.

By the way: I understand you are using Upp for 64 bit windows development.
Since I got my new machine, yesterday, I wonder if the conversion from 32 bit is a smooth ride.

I guess it always was. Well, before last batch of changes in debugger, you had to use 32-bit theide and was only able to debug 32-bit apps (but you could compile 64-bits even then). Now you can use 32-bit compiled theide.exe in the same mode as before and 64-bit for both 32-bit and

64-bit debugging.

Are you using Visual Studio 13?
[/quote]

I am only using the free SDK, usually, it includes all you need for U++.

Mirek

Subject: Re: Problems debugging with Visual C++
Posted by cbpporter on Fri, 19 Sep 2014 09:38:00 GMT
View Forum Message <> Reply to Message

I can confirm that at least in my case the recent issues with debugging are fixed and debugging in general is better that it was with a few months old build under 32 bit.

Thank you!