Subject: Arjun Bijanki: Making Sense of VC Intellisense
Posted by unodgs on Fri, 01 Feb 2008 13:15:59 GMT
View Forum Message <> Reply to Message

..quite interesting speach about how MS does inellisense
http://channel9.msdn.com/showpost.aspx?postid=376182

Subject: Re: Arjun Bijanki: Making Sense of VC Intellisense
Posted by mirek on Fri, 01 Feb 2008 18:56:34 GMT
View Forum Message <> Reply to Message

Anything relevant/interesting there? I do not have guts to spend 35 minutes watching a video...

Mirek

Subject: Re: Arjun Bijanki: Making Sense of VC Intellisense
Posted by mr_ped on Sat, 02 Feb 2008 01:54:16 GMT
View Forum Message <> Reply to Message

extra short summary:
old VS:
compiler has parser (perfect one)
intellisense has parser (not complete)
2 parsers baaaad
performance baaad, sometimes during writing you don't get characters drawn as the intellisense is working hard

new VS:
let's use compiler's (perfect) parser, right?
just the output of compiler will be not object file, but intellisense database.
1 parser = goood
Oh, and by the way we will mention we have now got two de facto compilers, one real, one intellisense DB producer, but having two compilers can't be bad, even if we publicly call them "frankenstain compiler" in video.
And the whole crap runs in background, so performance has been improved a lot.
And some fancy speech about united codebase/etc.

My own summary:
I don't see any major change except writing it from scratch. I think pointing out you are using compiler's parser instead of it's own is just giving thing different name. There was no limit to put compiler's parser in previous model. Also the performance improvement ... I'm skeptical ... running in background = huh, that should have been there since the very start.

Still VS is based on the old text processor and text source files philosophy, which is obsolete right now, and unless they ditch the whole main idea behind VS, they can only redo all the feature

again and again with better and better implementation, but I don't expect any major breakthrough there.
But of course it's always nice to get better implementation of the same stuff.

To TheIDE it's relevant a lot .. and yet not. As TheIDE is based on the same concept as VS, everything from video applies, but ... yeah, we are not using our own compiler .. which makes ripping out the parser of it and change it to produce browser data a hefty task, actually very likely as bad as doing your own imperfect parser with heuristics.

Still I wonder when we will see finally some full featured powerful IDE based on the full DB concept, and I mean not only simulating classic sourcefiles by storing code both in DB and producing files, but finally baning source files completely (let them survive in "Export sources") and using interactive pre-compilation of code not only to generate browser info, but also to produce real-time interactive statistics about performance, results for test parameters, etc... And producing full binary only on Run request with using precompiled pieces of stuff..... ... ...
Alright, I'm going rather to dream into bed, but I hope one day somebody will do it right.
It's still cutting out only "accident" things (essay Essence vs Accident tasks in programming from The Mythical Man-Month, if you don't have idea what I'm talking about), but I think the current work of programmer is still cluttered too much by them and worth of that final blow to them to get rid of them almost completely and focus on essence.
Also the DB based IDE with proper parsing could help attacking Essence itself a little bit, especially with last 5 years development of things like google-source and lot of matured OSS on internet and well described patterns some powerful search heuristic would be probably able to suggest very usable options for your //TODO comments.

Subject: Re: Arjun Bijanki: Making Sense of VC Intellisense
Posted by mirek on Sat, 02 Feb 2008 08:10:08 GMT
View Forum Message <> Reply to Message

mr_ped wrote on Fri, 01 February 2008 20:54Still I wonder when we will see finally some full featured powerful IDE based on the full DB concept, and I mean not only simulating classic sourcefiles by storing code both in DB and producing files, but finally baning source files completely (let them survive in "Export sources") and using interactive pre-compilation of code not only to generate browser info, but also to produce real-time interactive statistics about performance, results for test parameters, etc... And producing full binary only on Run request with using precompiled pieces of stuff..... ... ...

Well, I was thinking about it too and I even believe that this is what IBM environment does.... Still I am afraid this idea somewhat kills some C++ features which make it so appealing language. E.g. how .iml / .lay concepts fit into this image?

I know, these things could be integrated, but the C++ magic is that any application can develop something similar to cover its problem domain. This is what makes C++ flexible.

Therefore I am afraid that in order to preserve these favorable things, we will have to find the right degree of symbiosis between classical sources and "class" database access...

Mirek

Subject: Re: Arjun Bijanki: Making Sense of VC Intellisense
Posted by mr_ped on Sat, 02 Feb 2008 20:44:00 GMT
View Forum Message <> Reply to Message

How does the .iml and .lay relate to this?
I hope I'm not wrong, but both .iml and .lay files are generated by TheIDE. And they are still valid C++ sources, aren't they?

So such new IDE would be used to edit layout too, and it would generate the content of "lay" files too, and it would handle the "files" problems for them too (hopefully in completely transparent way to programmer).

Still the (continuous) import of external (generated or not) source files would be needed of course for many projects, but I think this can be worked out too without making it more difficult than with current IDEs, if not easier.

After all to attack "accident" part of programming means to make the life of programmer easier, not harder. Whether in classic source files based IDE or some future tool based on different concept doesn't matter. As long as the life is *really* easier.

(That means I'm really looking for improved TheIDE too, I bet it will make the life easier. I'm not just waiting for final miracle, any little improvement is welcome.)

Subject: Re: Arjun Bijanki: Making Sense of VC Intellisense
Posted by Mindtraveller on Sun, 03 Feb 2008 03:17:20 GMT
View Forum Message <> Reply to Message

I`m afraid there are two major points in here:
1) Real compiler (MSC, GCC) is very much "heavier" than integrated parser. So you`ll have your system performance degraded, especially on old PCs - even while editing the code.
2) Real compiler creates binary output only for "complete", non-error source file. So while editing the code, 80% of time this code is incomplete and has potential errors. I don`t know how integrated parser works, but I think it may be more flexible than keep-last-good-build approach.