Subject: Great (and funny) Linus' speach about GIT
Posted by unodgs on Tue, 18 Dec 2007 21:48:41 GMT
View Forum Message <> Reply to Message

http://pl.youtube.com/watch?v=4XpnKHJAok8

Subject: Re: Great (and funny) Linus' speach about GIT
Posted by mr_ped on Wed, 19 Dec 2007 12:59:56 GMT
View Forum Message <> Reply to Message

Impressive. And IMHO not really funny.
It's more sad than funny.

I'm used to SVN and very much like it, but that's mostly because I have full control over my own repository and because the number of people committing is limited and could be managed in decent time.
But I see where his hate for CVS-like is coming from and he has some very good points.

I will have to take some closer look on GIT after this, to get some real experience about different approach to code versioning.

Thanks for the link.

Subject: Re: Great (and funny) Linus' speach about GIT
Posted by fudadmin on Wed, 09 Jan 2008 18:09:17 GMT
View Forum Message <> Reply to Message

hi,
First of all, happy New Year to all.
Once again sorry for my lengthy disappearance, part of which was my "ugly stupidity" (but even stronger as in Linus's speech) of not having a proper system of backups and version control. I've never imagined before that hard disks can malfunction and, eventually, die so frequently...
Positive result, though, I've got a new PC (HP T7200 2Gb 2Ghz Core Duo tablet (...until Apple starts releasing tablets or I'll use my experience to hackintosh (I've got Apple licence)) which is able to rebuild all upp ide in ~30sec(!) instead of ~40 min 256MB Athlon 2800+.
The great thing about Linus's speech (thanks, uno, for the link!) that it opened my eyes or, actually, confirmed my gut feelings, that the notion "distributed" will dominate the world for the next coming years or even decades. Autocracy and related paradigms must die. "God" and "Central" is against the nature and evolution. (God is God when distributed?). Web 3.0 should reflect those changes. Wikipedia (as it is now) and centralized wikis should die. Progress and evolution can be most effective amongst groups of peers.
Conclusion and future dreams:
1. uppsrc under DVC (Distributed Version Control) (I'll definetely make my branches public after I make order with my home and hosted networks). Anyone else has done already (apart from Mirek's) or wants to follow?

2. Topic++ and forums connected, under DVC and accesible from theIde (does anyone want to investigate Rebol?)

Some technical (political?) questions:
1. Why uvs2 is not shipped with upp installation package, not on upp sourceforge, not on U++ webpages (or am I blind?) ?
2. how much does uvs2 compare to git (or needs changes to be better)?
3. what would be needed to use uvs2 for a "distributed system" (only setup FTP servers and exchange settings)?

Regards, Aris

Subject: Re: Great (and funny) Linus' speach about GIT
Posted by mirek on Wed, 09 Jan 2008 20:58:31 GMT
View Forum Message <> Reply to Message

fudadmin wrote on Wed, 09 January 2008 13:09
1. Why uvs2 is not shipped with upp installation package, not on upp sourceforge, not on U++ webpages (or am I blind?) ?

Because we are feeling guilty about it...

Quote:
2. how much does uvs2 compare to git (or needs changes to be better)?

Not much. Solves problem of sharing sources between a low number of very active developers that only have ftp server...

Quote:
3. what would be needed to use uvs2 for a "distributed system" (only setup FTP servers and exchange settings)?

A start from the scratch IMO.

Mirek

P.S.: Welcome back again

Subject: Re: Great (and funny) Linus' speach about GIT
Posted by fudadmin on Thu, 10 Jan 2008 00:31:20 GMT
View Forum Message <> Reply to Message

luzr wrote on Wed, 09 January 2008 20:58fudadmin wrote on Wed, 09 January 2008 13:09
1. Why uvs2 is not shipped with upp installation package, not on upp sourceforge, not on U++ webpages (or am I blind?) ?

Because we are feeling guilty about it...

1. Is that guilt more important than a chance to be useful for existing and new U++ users?
2. Is it that bad that ... but why then you are using on a grand scale for U++ yourself?
3. Why no other version control system is recommended on the website?
4. Guilty of what? Comunist system inheritance? I still remember uvs2 from my short experience that it is nicely useable with theIde. A product which potentially could promote U++ or at least to be appreciated by "ugly stupid uneducated" U++ users like me, or to be a starting point for new versions developements.
And, honestly, I stopped using it after similarly "guilty ugly" Mirek's comments somewhere in these forums.

OTOH, the second great point and observation from Linus's speech is that... If Mirek had at least 10% of Linus's "positively ugly agressive dictionary"..., U++ would be in front of "ugly fatty stupid wxWidgets" and not "unknown, deleted and once again scheduled for deletion from "half-porn" origins Wikipedia -
http://business.timesonline.co.uk/tol/business/industry_sect ors/media/article774973.ece. But that's separate topics.

So, any chance to overcome that guilt and put uvs2 into e.g sourceforge uppdev tree?

luzr wrote on Wed, 09 January 2008 20:58fudadmin wrote on Wed, 09 January 2008 13:09
2. how much does uvs2 compare to git (or needs changes to be better)?

Not much. Solves problem of sharing sources between a low number of very active developers that only have ftp server...

1. What would be needed to improve the situation?
2. Shell I offer an ftp server with 10 accounts?

luzr wrote on Wed, 09 January 2008 20:58fudadmin wrote on Wed, 09 January 2008 13:09
3. what would be needed to use uvs2 for a "distributed system" (only setup FTP servers and exchange settings)?

A start from the scratch IMO.

1. 2 weeks as for Linus?
2. What if to combine uvs2(3?) (theIde?, too?) and mercurial.
3. I'm starting to invest some time into uvs2 (first of all finding...) and/or mercurial (already installed it), anyway.

4. Depending on your answers and community response I'll start new topics in appropriate sections. I want order with my sources!

Quote:
P.S.: Welcome back again

Really appreciate that.
Aris

Subject: Re: Great (and funny) Linus' speach about GIT
Posted by fudadmin on Thu, 10 Jan 2008 06:54:04 GMT
View Forum Message <> Reply to Message

ok, I've found old uvs2 (v2.36). Before syncing I'll try to read forums (already noticed some topics) and docs (if any) and then ask some more questions...

Subject: Re: Great (and funny) Linus' speach about GIT
Posted by mdelfede on Thu, 10 Jan 2008 11:16:06 GMT
View Forum Message <> Reply to Message

fudadmin wrote on Thu, 10 January 2008 01:31
2. Is it that bad that ... but why then you are using on a grand scale for U++ yourself?

from what I've seen on UVS2, it makes it difficult to go back through versions (maybe I'm wrong, but I think so). So, if you post wrong code, it can be time consuming to sync it again.
Quote:
3. Why no other version control system is recommended on the website?

Well, now there's svn updated quite often, if you need to get the last code. A development trunk for external contributions could be opened and kept in sync by somebody having write access to uvs2 repository. I'd keep the main svn trunk exactly in sync with uvs one, as is it now, but I see no problem to open another branch.
Of course, that'll make a bit difficult to keep all in sync, but can be done.
The only real problem I see is that, with sourceforge svn you can't open a single branch for write access (IMO), so who has write access to the development branch has write access to main branc too.
Quote:
.....
1. What would be needed to improve the situation?

a miracle ?
Seriously speaking, as Mirek told, I see UVS as a mean for very *few* active developers to share code. As is it done, with many developers with write access it could be cumbersome to mantain.
Quote:
2. What if to combine uvs2(3?) (theIde?, too?) and mercurial.

I've looked inside mercurial for a while and it seems good stuff, but it has (IMHO) the weakness of all distributed versionin control systems. It relies mostly on active code shares between developers. AND, it doesn't have a merge tool embedded.
So, if 50 developers work on it, that'll be 50 different repositories with must be kept in sync. So, you'd need again a central FTP server (IMHO...) for repository sharing. OR you need the 50 developers share themselves the repo.
Linux kernel is a different stuff, it has a central developer (Linus) that join all the patches together, accepting or rejecting them. In my opinion, that's not a true decentralized system, even if it uses a decentralized versioning system.

Concluding.... I'm still thinking that a good centralized repo like svn works better for upp purpose (please, no flames about !)

Ciao

Max

Subject: Re: Great (and funny) Linus' speach about GIT
Posted by mr_ped on Thu, 10 Jan 2008 13:04:55 GMT
View Forum Message <> Reply to Message

I quite agree with you about the need of central point.
Than again I quite agree with Linus about cumbersome merges in CVS and SVN.
Did you notice that if you try to merge the same commit from other branch two times, the SVN goes hairwire instead of ignoring it?
Yes, it's user mistake to try to merge the same changes two times over code base, but the practical implications do result into "strong opinion".

So if there is free versioning system with better merge tools than SVN+TortoiseSVN (the tortoise merge is not bad to see changes, but far from "the" artificially intelligent merge tool which would do much of (obvious) merge work for you - I would go for it.

I didn't try GIT yet, so I'm not sure how smooth the merges are there. IF they are really as nice as Linus advocates, it would be probably easier to force Mirek be a "Linus" for upp and having his repository as "central" one, than to fight with SVN later, if the user base grows (and I think it will grow.. maybe not fast, but so far every year there are more people here using upp).

I think the basic choice about centralized vs distributed is about how many contributors to core upp we expect in future. Whether Mirek wants to keep it as his personal piece of SW with few core developers, or he wants to release upp into wild and let everybody do with it whatever he wants, and than upon popular request merging those improvements back into official version.
(well, the upp license already allows anybody to take the sources and do whatever they wish with them, but the process of propagating those changes back to official release is not defined at all, and works on good taste+political skills in forums+mood of Mirek or Unodogs. This view is probably not very encouraging for new developers to [try to] contribute directly to core.)

---

mr_ped wrote on Thu, 10 January 2008 14:04I quite agree with you about the need of central point.
Than again I quite agree with Linus about cumbersome merges in CVS and SVN.
Did you notice that if you try to merge the same commit from other branch two times, the SVN goes hairwire instead of ignoring it?

No, I didn't but I can imagine that
Well, outside svn I use 'meld', but it works only on Linux. It's a really good diff/merge program.

Quote:
Yes, it's user mistake to try to merge the same changes two times over code base, but the practical implications do result into "strong opinion".

yes, such mistakes are very common.... svn should behave better on them.

Quote:
So if there is free versioning system with better merge tools than SVN+TortoiseSVN (the tortoise merge is not bad to see changes, but far from "the" artificially intelligent merge tool which would do much of (obvious) merge work for you - I would go for it.

Me too. I'd like a centralized versioning system with a really good merge tool, but there isn't any yet. But it can be done as I do, with SVN and an external merge tool. Not too confortable, bt better than nothing...

Quote:
I didn't try GIT yet, so I'm not sure how smooth the merges are there. IF they are really as nice as Linus advocates, it would be probably easier to force Mirek be a "Linus" for upp and having his repository as "central" one, than to fight with SVN later, if the user base grows (and I think it will grow.. maybe not fast, but so far every year there are more people here using upp).

I think the basic choice about centralized vs distributed is about how many contributors to core upp we expect in future. Whether Mirek wants to keep it as his personal piece of SW with few core developers, or he wants to release upp into wild and let everybody do with it whatever he wants, and than upon popular request merging those improvements back into official version.

I'm in favour of a centralized system because with a decentralized one it's too easy that a developer forget to sync/merge to other's repos. Keeping a centralized ftp repo with mercurial would be better, but it depends everytime of the syncs from each developer. Which has to sync their repos, then ftp them on central site. I'm afraid of what would happen if 2 developers transfer at the same time their repos on ftp....

I guess for the moment the best would be a svn repo with a main 'stable' trunk, updated only by Mirek or a few developers, and some development branches to accept contibutions.
But that one can't be on sourceforge *or* you must accept the risk of giving write access to the full svn repo to everybody.

Ciao

Max

Subject: Re: Great (and funny) Linus' speach about GIT
Posted by mirek on Thu, 10 Jan 2008 16:30:14 GMT
View Forum Message <> Reply to Message

fudadmin wrote on Wed, 09 January 2008 19:31luzr wrote on Wed, 09 January 2008 20:58fudadmin wrote on Wed, 09 January 2008 13:09
1. Why uvs2 is not shipped with upp installation package, not on upp sourceforge, not on U++ webpages (or am I blind?) ?

Because we are feeling guilty about it...

4. Guilty of what? Comunist system inheritance? I still remember uvs2 from my short experience that it is nicely useable with theIde. A product which potentially could promote U++ or at least to be appreciated by "ugly stupid uneducated" U++ users like me, or to be a starting point for new versions developements.

Well, I have been told that Uvs2 is a bad idea so many times that I started to believe it

Anyway, reason 2 why it is not "public" is limited bandwidth of my home ftp server.

Mirek

Quote:
U++ would be in front of "ugly fatty stupid wxWidgets" and not "unknown, deleted and once again scheduled for deletion from "half-porn" origins Wikipedia - http://business.timesonline.co.uk/tol/business/industry_sect ors/media/article774973.ece. But that's separate topics.

Already tried that. Does not work.

Quote:
So, any chance to overcome that guilt and put uvs2 into e.g sourceforge uppdev tree?

It is in uppbox (together with other developer tools, like releaser).

Quote:
2. What if to combine uvs2(3?) (theIde?, too?) and mercurial.

One suggested plan was to make uvs3 as smart frontend to svn.

Mirek

Subject: Re: Great (and funny) Linus' speach about GIT
Posted by mirek on Thu, 10 Jan 2008 16:36:10 GMT
View Forum Message <> Reply to Message

mr_ped wrote on Thu, 10 January 2008 08:04Whether Mirek wants to keep it as his personal piece of SW with few core developers, or he wants to release upp into wild and let everybody do with it whatever he wants, and than upon popular request merging those improvements back into official version.
(well, the upp license already allows anybody to take the sources and do whatever they wish with them, but the process of propagating those changes back to official release is not defined at all, and works on good taste+political skills in forums+mood of Mirek or Unodogs. This view is probably not very encouraging for new developers to [try to] contribute directly to core.)

Well, you have to take into consideration that we use U++ for bussines, including some quite critical code. We have to do all we can to avoid big failures.

Anyway, "good taste+political skills in forums+mood of Mirek or Unodogs", does not sound too bad to me (perhaps with the exception of "mood"). Was not Linux developed in the same way?

Mirek

Subject: Re: Great (and funny) Linus' speach about GIT
Posted by fudadmin on Thu, 10 Jan 2008 17:14:21 GMT
View Forum Message <> Reply to Message

Big thanks for your very informative answers.
[quote title=mdelfede wrote on Thu, 10 January 2008 11:16
...The only real problem I see is that, with sourceforge svn you can't open a single branch for write access (IMO), so who has write access to the development branch has write access to main branch too
...
Max[/quote]
etc. etc.

mr_ped wrote
...and works on good taste+political skills in forums+mood of Mirek or Unodogs. This view is probably not very encouraging for new developers to [try to] contribute directly to core.)

That means, I'm not alone who noticed that...

So, branches are discouraged "not only by political but also by technical centralized autocratic upp project management nature and theIde version control unfriendliness"? like during good Soviet times

So, only not very skilled users are welcome.

Because skilled ones, after facing U++'s theIde and libraries quite huge limitations will want to adjust it for their own needs and contribute, but instead, will be forced to waste time going through politics (as in most projects), also own branches maintenance hell due to theIde nature (which even can't follow symlinks and MS shortcuts as a possible mechanism for switching between those branches), or silently watch and wait, or even leave (where is hojtsy e.g?)?

Imagine an impact on U++ populiarity if it had its own made that versioning-merging tool we and many world programmers are all dreaming about!

Conclusions, or back to reality from all of yours points:
1. I can try (starting tonight) to create a separate public subversion repository with write access branches for all wanting to share contributions U++ friends, if
 A. you really think that mercurial can't be used for a central repository role (and Linus is wrong, as he is about C++...)
 B. Mirek or others can't offer any better alternatives soon.

What do you think?

P.S I'm convinced that a fork of U++ would be a huge waste of small resources, not comparable to a Mozart of C++ (Mirek) and should be forgotten until Mirek is alive

---

Subject: Re: Great (and funny) Linus' speach about GIT
Posted by mdelfede on Thu, 10 Jan 2008 17:36:53 GMT
View Forum Message <> Reply to Message

fudadmin wrote on Thu, 10 January 2008 18:14Big thanks for your very informative answers.
mdelfede wrote on Thu, 10 January 2008 11:16
...The only real problem I see is that, with sourceforge svn you can't open a single branch for write access (IMO), so who has write access to the development branch has write access to main branch too
...
Max
etc. etc.

you're wellcome

Quote:

That means, I'm not alone who noticed that...

So, branches are discouraged "not only by political but also by technical centralized autocratic upp project management nature and theIde version control unfriendliness"? like during good Soviet times
ecc ecc....

You should maybe think also that theide (and upp) is quite stable because of its controlled development cycle.
Nobody is telling you that you can't share your patches and/or send to developers.... what you can't to is to fiddle directly with main code repo.
I must say that, being upp used by commercial development by Mirek & C, I agree fully with their restricted access to main repo. That's what, more or less, is doing Linus with linux kernel, too.
Besides of last year 2-3 crashes I had in Linux version (due mostly to lack of testing on Linux), I run daily theide (last uvs version) with absolutel *no* crashes at all.
I tried kdevelop times ago, and it was not half as stable as theide.

Quote:
Imagine an impact on U++ populiarity if it had its own made that versioning-merging tool we and many world programmers are all dreaming about!

and imagine the loss of popularity of U++ if theide would hang every 3 minutes due to badly tested code patches....

Quote:
Conclusions, or back to reality from all of yours points:
1. I can try (starting tonight) to create a separate public subversion repository with write access branches for all wanting to share contributions U++ friends....

that would be a great idea, IF you could keep that repo in sync with uvs development, along with users provided patches... if you have a lot of spare time, that'll really great stuff

Ciao

Max

Subject: Re: Great (and funny) Linus' speach about GIT
Posted by fudadmin on Thu, 10 Jan 2008 17:48:43 GMT
View Forum Message <> Reply to Message

Quote:It is in uppbox (together with other developer tools, like releaser).

from https://upp.svn.sourceforge.net/svnroot/upp I only get
======
/.svn/

/bazaar/
/examples/
/reference/
/tutorial/
/uppsrc/
=======
Any ideas?

Subject: Re: Great (and funny) Linus' speach about GIT
Posted by zsolt on Thu, 10 Jan 2008 18:11:15 GMT
View Forum Message <> Reply to Message

fudadmin wrote on Thu, 10 January 2008 18:14

So, branches are discouraged "not only by political but also by technical centralized autocratic upp project management nature and theIde version control unfriendliness"? like during good Soviet times

So, only not very skilled users are welcome.

I can not agree...

I think, the main advantages of U++ are it's very clean design and clean source code.
The result of allowing a lot of developers writing the repositry can be a hacked unstable trash. Just compare the development methods of some died projects to some flourishing projects.

You can write clean code, fitting into the concepts of UPP and it can be part of it (my observation).

OTOH, it would be benefical to create some infrastructure to maintain a list of contributed modules (packages).

Subject: Re: Great (and funny) Linus' speach about GIT
Posted by mdelfede on Thu, 10 Jan 2008 18:18:20 GMT
View Forum Message <> Reply to Message

zsolt wrote on Thu, 10 January 2008 19:11
OTOH, it would be benefical to create some infrastructure to maintain a list of contributed modules (packages).

Here I strongly agree. A common place where tu put patches and code snippets/add-on would be great.
Maybe some http stuff where people can upload the code in zip format along with a description of

what it is.
Bazaar in svn is not enough, IMHO.

Ciao

Max

Subject: Re: Great (and funny) Linus' speach about GIT
Posted by fudadmin on Thu, 10 Jan 2008 18:31:59 GMT
View Forum Message <> Reply to Message

zsolt wrote on Thu, 10 January 2008 18:11fudadmin wrote on Thu, 10 January 2008 18:14

So, branches are discouraged "not only by political but also by technical centralized autocratic upp project management nature and theIde version control unfriendliness"? like during good Soviet times

So, only not very skilled users are welcome.

I can not agree...

I think, the main advantages of U++ are it's very clean design and clean source code.
The result of allowing a lot of developers writing the repositry can be a hacked unstable trash. Just compare the development methods of some died projects to some flourishing projects.

You can write clean code, fitting into the concepts of UPP and it can be part of it (my observation).

I'm not talking about "to write to the main (Mirek's) branch".
I'm talking about a mechanism which whould enable quite average c++ programmers to be encouraged and empowered to contribute and, more importantly, to exchange U++ patches or branches in a more effective way than to post and search on forums!!!

Subject: Re: Great (and funny) Linus' speach about GIT
Posted by zsolt on Thu, 10 Jan 2008 18:36:11 GMT
View Forum Message <> Reply to Message

I did'n know about your Bazaar repo
Bazaar will be very useful, if you or somebody else maintain and regularly release it, and publish its content with detailed descriptions, I think.

Subject: Re: Great (and funny) Linus' speach about GIT
Posted by mr_ped on Thu, 10 Jan 2008 20:04:23 GMT
View Forum Message <> Reply to Message

- what's wrong with whole svn writing rights, if the agreement will be to not commit any big change to official branch?
If anyone does commit something there, it can be rolled back, eventually he can lose svn access.
Should we be so hard even on obvious fixes of obvious defects? I would not like such policy. But if they will be allowed, who will be responsible for review of commit and accepting it as official fix?
Also if Mirek will continue to work with uvs as main branch, who will merge those two "official" branches?
Also if Mirek will start to work directly on official svn branch, it may become easily unstable for couple of days ...

- stable upp core because of Mirek doing biz apps with it: that's obvious reason.

- good taste+forum politics = I was mostly pointing out the process is unofficial. Thus in similar situation it may easily lead to different results and it would be more encouraging if there would be official how-to for contributors (even if it would just describe this method).

- I think uvs2 is bad idea, because it is non-standard, and works only on limited bandwidth for limited number of people. So I think the migration to different development model would be a step forward.
But it's up to Mirek to decide which direction that forward should take. SVN? GIT? Mercurial? Bazaar (versioning system)?
Plenty of choices, each does promise a bit different future, and require quite different policies and rules.

Forking upp core doesn't look very promising to me. I can hardly imagine anyone else pushing so hard forward and so effectively as Mirek is. That means any fork would become quite obsolete within weeks, and exactly because of this evergoing big changes/refactoring to core it's difficult to maintain custom core changes if they are not merged back to official branch.

(and also core is quite complete for what it does, there's still plenty of room for improvements, but so far I didn't see people here moaning about wrong changes to core from Mirek. About missing features and fixing bugs.. those are common in forums... but the new version of core were so far very positively accepted and it looks Mirek's needs follow very good needs of others)

Forking TheIDE makes much more sense IMHO, as people may have quite different taste for what is important for IDE, and also experimenting a lot on such piece of SW makes more sense, than some crazy experiment changes to core itself.
But actually I think it's not worth of it. If there will be enough courage to do this, I would rather suggest to start TheIDE2 from scratch, as the original IDE is based on (IMHO obsolete) old concepts anyway and just "hacking it" would never move it that much away from current thing.

--- I see I'm writing lot of stuff, and quite offtopic.

So short summary what I don't like about current svn<->uvs and contributing.
I don't like the idea that if I find some obvious bug in upp (and I already did at least once IIRC), I

can't simply fix and commit it. I have to write it in forum ... wait for Mirek to notice ... wait for next official release ... hope Mirek did not forgot to merge with my fix ... etc.
The current status is pretty much "Mirek is developing his UPP, and as he is very good person, he allows anyone else to download it and use it for free" (what is absolutely amazing of course).
As such the number of direct contributors to upp core will never grow too much.

I'm not saying we need to change this badly. I'm just summarizing how I see it is, and where it will lead in future.
If it stays as it is, I'm not going to fork upp core just to allow me to fix little things immediately, I will rather post in forums and wait.

Subject: Re: Great (and funny) Linus' speach about GIT
Posted by mdelfede on Thu, 10 Jan 2008 20:34:15 GMT
View Forum Message <> Reply to Message

mr_ped wrote on Thu, 10 January 2008 21:04- what's wrong with whole svn writing rights, if the agreement will be to not commit any big change to official branch?
If anyone does commit something there, it can be rolled back, eventually he can lose svn access.

You don't take in account the possible mistakes on commits, for example. The best would be the possibility for everyone to create a branch and work on it, but avoiding the public write access to other branches. Then, it would be easy for Mirek (or some mantainer) to fetch the working patches and to merge them to the main three (or even uvs2).

Quote:
Should we be so hard even on obvious fixes of obvious defects? I would not like such policy.

I think that 'obvious' fixes are 2-3 lines of code, max. And those would be easily handled in forums. Of course, a forum dedicated to patches would be good.
Less obvious fixes, that maybe can change core behaviour, can be more dangerous. Just look on the time it took Mirek to accept my X11 DHctrl control, which involved many small changes in core code... and even, when he accepted it and merged it still contained a pair of nasty bugs. He was right to be worried about fiddling with core classes, I must say. Just think what could happen if anybody could fiddle with other's threes.

I'm quite happy with upp and theide as is it now.
The small caveats it has are being solved quickly, and I see the reaction against bugs are quick enough.
Maybe an added mercurial repository for code experiments on a centralized ftp server would be a good stuff, but then it needs a mantainer to keep it in sync with uvs2 as I'm trying to do now with svn repo. It's not an hard job, but it needs a person that can do it almost daily, to be effective.

If there's a volunteer here to do it, it's not too hard to find a free ftp server, build a mercurial repo synced with svn and open it to all.

Ciao

Max

Subject: Re: Great (and funny) Linus' speach about GIT
Posted by mirek on Thu, 10 Jan 2008 21:55:01 GMT
View Forum Message <> Reply to Message

mr_ped wrote on Thu, 10 January 2008 15:04
I don't like the idea that if I find some obvious bug in upp (and I already did at least once IIRC), I can't simply fix and commit it. I have to write it in forum ... wait for Mirek to notice ... wait for next official release ... hope Mirek did not forgot to merge with my fix ... etc.
The current status is pretty much "Mirek is developing his UPP, and as he is very good person, he allows anyone else to download it and use it for free" (what is absolutely amazing of course).
As such the number of direct contributors to upp core will never grow too much.

I am not the only one able to apply patches. Daniel and Tom can do it too and both do.

IMO, the long term solution is to allow more developers the write access. And IMO it will happen. OTOH, we are obviously very careful here...

Mirek

Subject: Re: Great (and funny) Linus' speach about GIT
Posted by mirek on Thu, 10 Jan 2008 21:58:53 GMT
View Forum Message <> Reply to Message

mr_ped wrote on Thu, 10 January 2008 15:04
- I think uvs2 is bad idea, because it is non-standard, and works only on limited bandwidth for limited number of people. So I think the migration to different development model would be a step forward.
But it's up to Mirek to decide which direction that forward should take. SVN? GIT? Mercurial? Bazaar (versioning system)?
Plenty of choices, each does promise a bit different future, and require quite different policies and rules.

Well, that is the problem. I accept that uvs2 is no longer a good solution, but right now the migration effort does not seem worth it, especially when the options are many and none perfect... I am afraid of going from one unoptimal solution to another. At least, with uvs2, I know what to expect. And from time to time, we can fix problems too

Mirek

Subject: Re: Great (and funny) Linus' speach about GIT
Posted by fudadmin on Fri, 11 Jan 2008 00:14:03 GMT
View Forum Message <> Reply to Message

mdelfede wrote on Thu, 10 January 2008 20:34
I'm quite happy with upp and theide as is it now.

I'm not. It's driving me mad. Core, CtrlCore, CtrlLib, RichEdit... Nearly all of them. Because if something somewhere is better I'd like to have it...
Things like messed up home directory with logs, usrlog files on Linux and BSD? Config files where? Cleaning configs? (Ok, I've noticed that yesterday's commit contains .upp/ in App.cpp but how long did it take? Why into core hardcoded dirs? Freedesktop standards configs? start HTML viewer on Posix? DirTree Ctrl? CodeEditor for other languages (including external syntax files)? Right click menus in theide CodeEditor? Switching root-user from theide and e.g using dummy pakages as smart directories for linux/bsd configs (usc macro Input auto parser is nice) and having Topic++ as smart documentation? Archlinux abs upp_aris package... UWord - RichEdit linking and not insertion of picture (including agg svg) files, HTML code generation for different standards, HTML parser - converter into qtf and backwards? Better stylesheets management and editing for websites design (connected with php templates)? Esc and macro extended funtions like "macro recorder-player", extended upt templates and their parameters entering, MenuCreator -templator package (I like that very much) , MacOS style agg docker-menu for Posix (including OpenBSD, and MirBSD), ByteCode compiler for interpreted languages, standalone Dialect interpreter with U++ GUI Ctrls, C++ OO gigabase database packages? Using theide for Symbian60- epoc Nokia? ...
Have you got all that? I have or had. Messy as hell. Most of those I had working "as a proof of concept" level. But then it comes catching up with new official versions and merging...
And then I think if Mirek is so effective and can write such a clean and practically bugless code and he is using Uvs2... IT must be good.

...
And then I know that there people who work on the same things. Woldn't be better to share such things even they are very small and not perfectly or (at all) finished?

mdelfede wrote on Thu, 10 January 2008 20:34
Maybe an added mercurial repository for code experiments on a centralized ftp server would be a good stuff, but then it needs a mantainer to keep it in sync with uvs2 as I'm trying to do now with svn repo. It's not an hard job, but it needs a person that can do it almost daily, to be effective.

If there's a volunteer here to do it, it's not too hard to find a free ftp server, build a mercurial repo synced with svn and open it to all.

Ciao

Max

I've got ftp server with 10 accounts 10gb/month bandw now. (In a few days or weeks could recover another, bigger one if needed)
I'm a voluteer who can do that job almost daily.

So mercurial repo?

Subject: Re: Great (and funny) Linus' speach about GIT
Posted by mirek on Fri, 11 Jan 2008 08:03:56 GMT
View Forum Message <> Reply to Message

fudadmin wrote on Thu, 10 January 2008 19:14
I'm not. It's driving me mad. Core, CtrlCore, CtrlLib, RichEdit... Nearly all of them. Because if something somewhere is better I'd like to have it...
Things like messed up home directory with logs, usrlog files on Linux and BSD? Config files where? Cleaning configs? (Ok, I've noticed that yesterday's commit contains .upp/ in App.cpp but how long did it take? Why into core hardcoded dirs? Freedesktop standards configs? start HTML viewer on Posix? DirTree Ctrl? CodeEditor for other languages (including external syntax files)? Right click menus in theide CodeEditor? Switching root-user from theide and e.g using dummy pakages as smart directories for linux/bsd configs (usc macro Input auto parser is nice) and having Topic++ as smart documentation? Archlinux abs upp_aris package... UWord - RichEdit linking and not insertion of picture (including agg svg) files, HTML code generation for different standards, HTML parser - converter into qtf and backwards? Better stylesheets management and editing for websites design (connected with php templates)? Esc and macro extended funtions like "macro recorder-player", extended upt templates and their parameters entering, MenuCreator -templator package (I like that very much) , MacOS style agg docker-menu for Posix (including OpenBSD, and MirBSD), ByteCode compiler for interpreted languages, standalone Dialect interpreter with U++ GUI Ctrls, C++ OO gigabase database packages? Using theide for Symbian60- epoc Nokia? ...

Aris, what makes you think that that you need anything to get these?

Why do not you just sit and start coding?

If you need more developers participate, fine, you have Bazaar SVN repo.

BTW, I am still waiting for a reasonable AGG package without a catch. Something I can download, unpack to 1-3 packages, move into my uppdev nest and start testing. Something like extended testcase. It would be helpful, if for nothing else, to bechmark the performance of new planned painting module...

Mirek

Subject: Re: Great (and funny) Linus' speach about GIT
Posted by mr_ped on Fri, 11 Jan 2008 08:43:02 GMT
View Forum Message <> Reply to Message

"Why do not you just sit and start coding?"

I thought he was listing things he already *did*.
(probably not clean enough to be merged back into official upp to just send you testing package)

... about mirrored svn, mercurial (and whatever you like, I think after the SVN is finally synced with uvs2 (great!), this is no more problem of UPP crew, anybody can create whatever repository likes, and sync with synced SVN ... souds ugly? It will be ugly. But there's not much more there can be done about it without radical change in upp core team, i.e. defining new philosophy/policies of maintaining upp project, choosing the tools and migrating from uvs there).

As I wrote above, Mirek tends to change core sources a lot, he definitely likes to refactor code anytime he doesn't feel happy about it. (that's great actually)

So whenever you fork from official upp, do your own changes on let's say private branch in svn, if it takes you more than 2-3 months, you are facing ugly task of merging back to recent upp, especially if you don't merge continuously trough those 2 months.
If you work on something quite independent, this is not a real deal, if you work on something like that X11 DHctrl, it will probably make the process of "finishing" package much harder, as you are trying to hit a moving target.
Merging continuously is something what makes your base for that experiment (potentially) unstable and (for sure) "moving", and while upp is usually quite stable after refactoring, you may stop feeling safe.. if you don't feel safe, you may start to look for (your) bugs in upp code .. you will lose time, confidence, and get worser code in the end.
(just read some basics about Test Driven Development and the reasoning behind it, most of the TDD advocates say the improved confidence improves everything... coding speed, code quality, you name it. From my limited experience with TDD I can only agree with that, the confidence and ability to focus on single problem had earthshaking effect on my laziness and productivity, which has been quite low recent years on my side)

Also keep in mind Mirek does not announce what he is working on, so many bigger inner refactorings happen pretty much overnight in uvs2 without any hint they will appear. Had you finished your package on evening, wanting to give it another test and morning and send to Mirek as finished? It may be well obsolete at morning.

All these apply big time only to upp core development itself, Mirek's notice about giving more (but carefully) write permissions will lead to "low number of core developers", just as I said before. This is not a horrible thing, after all the upp has been like this for years and it does progress forward, so it's certainly a working philosophy/policy.
But it may look like some personal project for years due to this (and not like a serious development platform ... and while upp has got it's share of problems if you want it to exactly fit your biz needs, Mirek's own living proves the platform *is* viable and serious for commercial use already).

If you are doing work outside of core, you are usually bound only to upp API, and while that one does change quite a lot too, you can usually "port" your obsolete source to fresh upp version within minutes or hours, so you can work on stable base, and merge with latests only sometimes. Especially if you work with old stable parts of API, you are unlikely to hit any of these things I write

above about.

Subject: Re: Great (and funny) Linus' speach about GIT
Posted by mdelfede on Fri, 11 Jan 2008 10:12:51 GMT
View Forum Message <> Reply to Message

mr_ped wrote on Fri, 11 January 2008 09:43
... about mirrored svn, mercurial (and whatever you like, I think after the SVN is finally synced with uvs2 (great!), this is no more problem of UPP crew, anybody can create whatever repository likes, and sync with synced SVN ... souds ugly? It will be ugly. But there's not much more there can be done about it without radical change in upp core team, i.e. defining new philosophy/policies of maintaining upp project, choosing the tools and migrating from uvs there).

That's right for people be in sync with latest devel, not for people wanting to contribute. But again that's an hard stuff.
I guess Mirek must choose between having a tight control of his sources OR opening it to more developers. And, as he uses upp to develop commercial apps (and, I guess, to earn money of them) his choice must be the first one.

Quote:
As I wrote above, Mirek tends to change core sources a lot, he definitely likes to refactor code anytime he doesn't feel happy about it. (that's great actually)

And that's needed to stay tuned with, for example, new stuffs like windows Vista (beurk!). That, of course, because of some lack of code modularity in upp, to some extent. But again I can understand Mirek, adding more code modularity would mean a major rewrite of upp.... I wouldn't do it.

Quote:
So whenever you fork from official upp, do your own changes on let's say private branch in svn, if it takes you more than 2-3 months, you are facing ugly task of merging back to recent upp, especially if you don't merge continuously trough those 2 months.
If you work on something quite independent, this is not a real deal, if you work on something like that X11 DHctrl, it will probably make the process of "finishing" package much harder, as you are trying to hit a moving target.

That's the real problem. Contributors get an hard work to sync with upp continuosly before changes are merged in main three.
But I don't see an easy solution to that, without loosing control on code quality.

I think that we must distinguish betweeen 2 cases :

1- Changes to upp core stuffs
2- Adding new functionalities

For the first point, you'd need a different developing model, with all kind of problems it brings. I'd leave that like it is now, asking Mirek to insert patches inside main three after extensive testing.

The second point would be much easier for external developers IF
some sort of plugin system would be added to upp.
I'd like very much such a system... a plugin system with a SKD to develop external add-ons.

Ciao

Max

Subject: Re: Great (and funny) Linus' speach about GIT
Posted by mirek on Fri, 11 Jan 2008 11:10:20 GMT
View Forum Message <> Reply to Message

mr_ped wrote on Fri, 11 January 2008 03:43
But it may look like some personal project for years due to this (and not like a serious development platform ... and while upp has got it's share of problems if you want it to exactly fit your biz needs, Mirek's own living proves the platform *is* viable and serious for commercial use already).

If you are doing work outside of core, you are usually bound only to upp API, and while that one does change quite a lot too, you can usually "port" your obsolete source to fresh upp version within minutes or hours, so you can work on stable base, and merge with latests only sometimes. Especially if you work with old stable parts of API, you are unlikely to hit any of these things I write above about.

Well, how do you think Qt is being developed?

Sure, there is a company behind instead of "core team", but other than that, I see not much difference.

(BTW, in fact, there is a "company" behind U++ too:

http://www.justice.cz/xqw/xervlet/insl/report?sysinf.vypis.C
EK=314384&sysinf.vypis.rozsah=aktualni&sysinf.@typ=t
ransformace&sysinf.@strana=report&sysinf.vypis.typ=X
HTML&sysinf.vypis.klic=d391b6b69b0227d4dd916e94deba582f&
amp;sysinf.spis.@oddil=C&sysinf.spis.@vlozka=59328&s
ysinf.spis.@soud=M%ECstsk%FDm%20soudem%20v%20Praze&sysin f.platnost=11.01.2008

)

Mirek

Subject: Re: Great (and funny) Linus' speach about GIT
Posted by mirek on Fri, 11 Jan 2008 11:21:30 GMT
View Forum Message <> Reply to Message

Quote:
(just read some basics about Test Driven Development and the reasoning behind it, most of the TDD advocates say the improved confidence improves everything... coding speed, code quality, you name it. From my limited experience with TDD I can only agree with that, the confidence and ability to focus on single problem had earthshaking effect on my laziness and productivity, which has been quite low recent years on my side)

Well, I am afraid that is nice and good as long as you do not have to write tests for GUI libraries...

I am in fact extensively using test driven development for Core. New superoptimal String and new memory allocator were born this way. There is even 'upptst' repository to concentrate automated tests.

Anyway, I see no simple way how to apply this methodology for CtrlCore. And that is where the most tricky U++ bugs are...

Mirek

Subject: Re: Great (and funny) Linus' speach about GIT
Posted by mirek on Fri, 11 Jan 2008 11:27:08 GMT
View Forum Message <> Reply to Message

mdelfede wrote on Fri, 11 January 2008 05:12
I think that we must distinguish betweeen 2 cases :

1- Changes to upp core stuffs
2- Adding new functionalities

For the first point, you'd need a different developing model, with all kind of problems it brings. I'd leave that like it is now, asking Mirek to insert patches inside main three after extensive testing.

The second point would be much easier for external developers IF
some sort of plugin system would be added to upp.
I'd like very much such a system... a plugin system with a SKD to develop external add-ons.

It is nice how spending a while with CtrlCore changes the view, is not it?

I agree 100% with you. I also think that second point is already in progress, in last .dev the bazaar was the part of release, although not very well integrated.

The U++ is designed to be very modular. There should be a little problem with extending it by adding packages.

However, one problem I have noticed over years is that quite often, somebody contributes some code and then goes silent for a long time (hi, Aris!). This makes things difficult to integrate into 'uppsrc' - we are working hard to maintain these core parts, no energy left to maintain extensions...

So Bazaar seems quite OK for now. There a is clear distinction between Bazaar and uppsrc; and nobody will make me responsible for that code And you can contribute via SVN too.

Mirek

Subject: Re: Great (and funny) Linus' speach about GIT
Posted by mdelfede on Fri, 11 Jan 2008 13:05:48 GMT
View Forum Message <> Reply to Message

luzr wrote on Fri, 11 January 2008 12:27mdelfede wrote on Fri, 11 January 2008 05:12
I think that we must distinguish betweeen 2 cases :

1- Changes to upp core stuffs
2- Adding new functionalities

For the first point, you'd need a different developing model, with all kind of problems it brings. I'd leave that like it is now, asking Mirek to insert patches inside main three after extensive testing.

The second point would be much easier for external developers IF
some sort of plugin system would be added to upp.
I'd like very much such a system... a plugin system with a SKD to develop external add-ons.

It is nice how spending a while with CtrlCore changes the view, is not it?

ehehehehehehe... I knew it before, but you know, I like to try stuffs by myself

Quote:
I agree 100% with you. I also think that second point is already in progress, in last .dev the bazaar was the part of release, although not very well integrated.

The U++ is designed to be very modular. There should be a little problem with extending it by adding packages.

With 'plugin system' I mean a dll/.so loader, a sdk and some documented plugin api, that will allow to write some plugins in form of dll/.so without the need of recompile theide (which is not a big concern) and without the need of updating the add-ons on each core change (which IS a big concern).

Quote:
However, one problem I have noticed over years is that quite often, somebody contributes some code and then goes silent for a long time (hi, Aris!). This makes things difficult to integrate into 'uppsrc' - we are working hard to maintain these core parts, no energy left to maintain extensions...

The 'plugin mechanics' would solve at least 80% of this problem, leaving to core developers just the mantainment of plugin sdk. As is it now you must integrate each extension in theide/core... that is time consuming.

Quote:
So Bazaar seems quite OK for now. There a is clear distinction between Bazaar and uppsrc; and nobody will make me responsible for that code And you can contribute via SVN too.

Besides of plugins, bazaar *would* be good having a separate svn with public access for it, so it could become a place where everyone could contribute without limitations. It would be interesting to find an alternate svn hosting place for it...

Ciao

Max

EDIT : back to core 'fiddling', for me it was *very* difficult because I had to try to understand how it works.... And, I must say it, classes/function names and (missing) comments didn't make my life easy !

Subject: Re: Great (and funny) Linus' speach about GIT
Posted by mirek on Fri, 11 Jan 2008 13:29:05 GMT
View Forum Message <> Reply to Message

mdelfede wrote on Fri, 11 January 2008 08:05luzr wrote on Fri, 11 January 2008 12:27mdelfede wrote on Fri, 11 January 2008 05:12
I think that we must distinguish betweeen 2 cases :

1- Changes to upp core stuffs
2- Adding new functionalities

For the first point, you'd need a different developing model, with all kind of problems it brings. I'd leave that like it is now, asking Mirek to insert patches inside main three after extensive testing.

The second point would be much easier for external developers IF
some sort of plugin system would be added to upp.
I'd like very much such a system... a plugin system with a SKD to develop external add-ons.

It is nice how spending a while with CtrlCore changes the view, is not it?

ehehehehehehe... I knew it before, but you know, I like to try stuffs by myself

Quote:
I agree 100% with you. I also think that second point is already in progress, in last .dev the bazaar was the part of release, although not very well integrated.

The U++ is designed to be very modular. There should be a little problem with extending it by adding packages.

With 'plugin system' I mean a dll/.so loader, a sdk and some documented plugin api, that will allow to write some plugins in form of dll/.so without the need of recompile theide (which is not a big concern) and without the need of updating the add-ons on each core change (which IS a big concern).

While plugin system would be fine (but is pretty difficult with C++ in fact), I do not see how it is going to fix the problem, or more precisily, how it is better than packages.

Mirek

Subject: Re: Great (and funny) Linus' speach about GIT
Posted by fudadmin on Fri, 11 Jan 2008 14:38:59 GMT
View Forum Message <> Reply to Message

[quote title=luzr wrote on Fri, 11 January 2008 11:27
...
However, one problem I have noticed over years is that quite often, somebody contributes some code and then goes silent for a long time (hi, Aris!).

Mirek
[/quote]

Maybe they were not entirely happy with something related?
Ok, I don't want to waste too much time analyzing and explaining all possible reasons...
But... One thing I maybe (or maybe not) regret that you provoked me to get involved with that EyeCare comparison article and I went deep into web things. Because I'm a perfectionist and I wanted that comparison to be like I imagine... In quirks mode. Something like yahoo widgets and for different browsers with splitters, charts, buttons etc. (one conclusion - all browsers are bad and slow but MS Internet Explorer is the fastest). Because, how otherwise "ugly stupid" (using Linus words) wxwidgets fans would grasp that 282 lines is much less than 838 (and mostly ?.? times shorter...) from this kind of interface
http://www.arilect.com/upp/articles/compare_EyeCare/index.ht m ... An improved variant with some really nice graphics design I incorporated into Xaraya's http://www.xaraya.org/ php

templates and wrote some "small" helpers in U++ for that kind of developement. And? Then 2 or even more months I spend recovering pieces from my hard disk (No 4 in 2 years). Etc., etc. Doesn't matter. I'll publish that improved thing one day, anyway (but don't ask me when).
1. What matters is that a developer can't rely on one computer! 2. It's too expensive not to have a separate file server or servers.
3. And to work with a messy version control or without any is a (... put your worst word here)

So - advice me - from where, what and where to install and how to keep up in synchro with uppdev and do merges with less pain. Then expect my contributions very soon.

Aris
P.S. And mr_ped and mdelfede are very clever.

Subject: Re: Great (and funny) Linus' speach about GIT
Posted by mdelfede on Fri, 11 Jan 2008 15:09:00 GMT
View Forum Message <> Reply to Message

luzr wrote on Fri, 11 January 2008 14:29

While plugin system would be fine (but is pretty difficult with C++ in fact), I do not see how it is going to fix the problem, or more precisily, how it is better than packages.

It isn't better, is just different.
Packages are *very* good for developers, plugins are the best for users.
With a package you must :
- drop source inside upp tree
- recompile theide
AND, not having a stable interface, usually you must also mantain the extension along upp changes.

With a plugin, you get a dll (or .so), you just drop it on a plugin folder and you have a ready to use extension.
AND, if you've got a stable and documented sdk, you must recompile plugin only on major ide changes (i.e. changes on binary format, compiler, ecc), which does a big difference, IMHO.
That was the cause of success of Borland products, with their components.
We spoke about that a time ago, about the way controls are made and drawn. Your choice of having ontrols as part of upp (and drawn by esc code on layout editor) is not bad, but then if you add a control you must recompile the full ide *and* write an esc code for layout editor. Double job....

About the problems of c++, I don't see any big caveat.
Of course, the plugin must be recompiled for every compiler choice, but that's true even with c if you change platform.
BTW, plugins are inside code::blocks and they work.

OTOH, if you're concerned about c++ problems, you could make a c interface both on theide side and on sdk side... that would open the plugin system to other languages too.

Ciao

Max

Subject: Re: Great (and funny) Linus' speach about GIT
Posted by zsolt on Fri, 11 Jan 2008 16:46:42 GMT
View Forum Message <> Reply to Message

mdelfede wrote on Fri, 11 January 2008 16:09..., but then if you add a control you must recompile the full ide *and* write an esc code for layout editor. Double job....

This is not true. You yust have to add package to yours.

Subject: Re: Great (and funny) Linus' speach about GIT
Posted by mirek on Fri, 11 Jan 2008 17:04:14 GMT
View Forum Message <> Reply to Message

mdelfede wrote on Fri, 11 January 2008 10:09
OTOH, if you're concerned about c++ problems, you could make a c interface both on theide side and on sdk side... that would open the plugin system to other languages too.

Actually, each time I start thinking in this direction, I end with plugin as separate process and using something like Corba or SOAP for interface...

Mirek

Subject: Re: Great (and funny) Linus' speach about GIT
Posted by mirek on Fri, 11 Jan 2008 17:06:39 GMT
View Forum Message <> Reply to Message

mdelfede wrote on Fri, 11 January 2008 10:09
but then if you add a control you must recompile the full ide *and* write an esc code for layout editor. Double job....

As Zsolt pointed out, this is incorrect. Plus, in .dll style components, you have to write some

"development time" code too. So the only difference is that in U++, you write "development" code as .usc script...

Mirek

Subject: Re: Great (and funny) Linus' speach about GIT
Posted by mdelfede on Fri, 11 Jan 2008 17:19:39 GMT
View Forum Message <> Reply to Message

luzr wrote on Fri, 11 January 2008 18:04mdelfede wrote on Fri, 11 January 2008 10:09
OTOH, if you're concerned about c++ problems, you could make a c interface both on theide side and on sdk side... that would open the plugin system to other languages too.

Actually, each time I start thinking in this direction, I end with plugin as separate process and using something like Corba or SOAP for interface...

Well, it seems to me a bit complicated for the scope, but....

Quote:
As Zsolt pointed out, this is incorrect. Plus, in .dll style components, you have to write some "development time" code too. So the only difference is that in U++, you write "development" code as .usc script...

well, that's true. But in .dll style you're forced to do if ( ), with esc code you end up with half finished components.
But again, as components are used only by developers, I agree that your way is not bad at all (of course, it doesn't allow closed source components).
For plugins I see it a bit different.

BTW, I've got a small question about writing an enhancement to theide.... I want to make a 'reformat code' menu point, using astyle. How (if it's simple to tell me...) should I handle the editor change and the undo stuff ? My code should read the full buffer as a string and write it again inside editor once formatted.

Ciao

Max

Subject: Re: Great (and funny) Linus' speach about GIT
Posted by mirek on Fri, 11 Jan 2008 17:37:38 GMT
View Forum Message <> Reply to Message

mdelfede wrote on Fri, 11 January 2008 12:19luzr wrote on Fri, 11 January 2008 18:04mdelfede wrote on Fri, 11 January 2008 10:09
OTOH, if you're concerned about c++ problems, you could make a c interface both on theide side and on sdk side... that would open the plugin system to other languages too.

Actually, each time I start thinking in this direction, I end with plugin as separate process and using something like Corba or SOAP for interface...

Well, it seems to me a bit complicated for the scope, but....

Quote:
As Zsolt pointed out, this is incorrect. Plus, in .dll style components, you have to write some "development time" code too. So the only difference is that in U++, you write "development" code as .usc script...

well, that's true. But in .dll style you're forced to do if ( ), with esc code you end up with half finished components.
But again, as components are used only by developers, I agree that your way is not bad at all (of course, it doesn't allow closed source components).
For plugins I see it a bit different.

BTW, I've got a small question about writing an enhancement to theide.... I want to make a 'reformat code' menu point, using astyle. How (if it's simple to tell me...) should I handle the editor change and the undo stuff ? My code should read the full buffer as a string and write it again inside editor once formatted.

Ciao

Max

Generally, undo works on pretty low level - there are only two basic operations supported by editor - insert and remove. Both store records to undo queue.

In your case it would mean that if you perform your replace as "remove" / "insert", it should work... (I guess you will want to support selection reformatting as well, so this is more than natural IMO).

Mirek

Subject: Re: Great (and funny) Linus' speach about GIT
Posted by mdelfede on Fri, 11 Jan 2008 17:44:24 GMT
View Forum Message <> Reply to Message

luzr wrote on Fri, 11 January 2008 18:37

Generally, undo works on pretty low level - there are only two basic operations supported by editor - insert and remove. Both store records to undo queue.

In your case it would mean that if you perform your replace as "remove" / "insert", it should work... (I guess you will want to support selection reformatting as well, so this is more than natural IMO).

So, to reformat all editor code I should do a 'remove' of a selection of full code, then insert a new one... ok. Is it possible to group 2 operations in 1 undo step ? So, if I undo the change, I don't have on first undo the code fully erased ?

Subject: Re: Great (and funny) Linus' speach about GIT
Posted by zsolt on Fri, 11 Jan 2008 18:03:08 GMT
View Forum Message <> Reply to Message

mdelfede wrote on Fri, 11 January 2008 18:19
But again, as components are used only by developers, I agree that your way is not bad at all (of course, it doesn't allow closed source components).

I think, this is not true as well. Just check MySql or PostgreSQL package. They are a thin layer on top of a DLL. You can always do the same, if you want to provide a closed source package. (Maybe this could be mentioned in a HOWTO.)

mdelfede wrote on Fri, 11 January 2008 18:19
BTW, I've got a small question about writing an enhancement to theide.... I want to make a 'reformat code' menu point, using astyle. How (if it's simple to tell me...) should I handle the editor change and the undo stuff ? My code should read the full buffer as a string and write it again inside editor once formatted.

This can be a good starting point for you to implement a source manipulator plugin interface. It could be very simple, as it could work on simple "C" string buffers.
The plugin interface would declare a .dll (or .so) in a header file something like:

```
extern "C"{
  const char* ManipulateSourcecode(const char*);
  void FreeWorkingBuffer();
  const char* GetPluginName();
}
```

Subject: Re: Great (and funny) Linus' speach about GIT

Posted by mirek on Fri, 11 Jan 2008 18:04:42 GMT
View Forum Message <> Reply to Message

mdelfede wrote on Fri, 11 January 2008 12:44luzr wrote on Fri, 11 January 2008 18:37

Generally, undo works on pretty low level - there are only two basic operations supported by editor - insert and remove. Both store records to undo queue.

In your case it would mean that if you perform your replace as "remove" / "insert", it should work... (I guess you will want to support selection reformatting as well, so this is more than natural IMO).

So, to reformat all editor code I should do a 'remove' of a selection of full code, then insert a new one... ok. Is it possible to group 2 operations in 1 undo step ? So, if I undo the change, I don't have on first undo the code fully erased ?

Yes, sorry, forgot to mention. Steps are grouped by "serial id". If you call NextUndo, you start another undo block. Actually, these things are even documented, see TextCtrl documentation.

Mirek

Subject: Re: Great (and funny) Linus' speach about GIT
Posted by fudadmin on Mon, 14 Jan 2008 03:55:00 GMT
View Forum Message <> Reply to Message

luzr wrote on Fri, 11 January 2008 08:03
...

Why do not you just sit and start coding?

If you need more developers participate, fine, you have Bazaar SVN repo.

BTW, I am still waiting for a reasonable AGG package without a catch. Something I can download, unpack to 1-3 packages, move into my uppdev nest and start testing. Something like extended testcase. It would be helpful, if for nothing else, to bechmark the performance of new planned painting module...

Mirek

Was the agg plugin package, which I subtmitted:
http://www.ultimatepp.org/forum/index.php?t=msg&&th= 2416&goto=10449#msg_10449
unreasonable or with a catch?

You never told what was wrong with it. And why it was never included into upp downloads?

Or to use something with Ultimate++ you must first narrow the api with Upp way of wrapping things?
Even for the latter goal, maybe more new users could have been attracted - if they found it and were able to use it in the original agg way - and then contribute to those wrappers?
You don't test win32 libs to be used with U++, do you?

Or if are concerned about the purity and quality of some packages, why not to add a subfolder "3rdParty" like other projects do?

P.S And if you write code practically without comments and documentation, and expect people to grasp everything in a few weeks and at the same time learn libraries like agg, write articles, do other things etc., and to be not frustrated when ...?

How many of those developers still participate from the very beginning?...

Subject: Re: Great (and funny) Linus' speach about GIT
Posted by mirek on Mon, 14 Jan 2008 10:36:01 GMT
View Forum Message <> Reply to Message

fudadmin wrote on Sun, 13 January 2008 22:55luzr wrote on Fri, 11 January 2008 08:03
...

Why do not you just sit and start coding?

If you need more developers participate, fine, you have Bazaar SVN repo.

BTW, I am still waiting for a reasonable AGG package without a catch. Something I can download, unpack to 1-3 packages, move into my uppdev nest and start testing. Something like extended testcase. It would be helpful, if for nothing else, to bechmark the performance of new planned painting module...

Mirek

Was the agg plugin package, which I subtmitted:
http://www.ultimatepp.org/forum/index.php?t=msg&&th= 2416&goto=10449#msg_10449
unreasonable or with a catch?

You never told what was wrong with it. And why it was never included into upp downloads?

Actually, it is a very good start.

But somehow I suspect users want to draw more than lines and ellipses...

Quote:
Or to use something with Ultimate++ you must first narrow the api with Upp way of wrapping

things?

Actually, yes. I would like to have more intuitive interface and better integrated with U++.

Quote:
Even for the latter goal, maybe more new users could have been attracted - if they found it and were able to use it in the original agg way - and then contribute to those wrappers?

Or if are concerned about the purity and quality of some packages, why not to add a subfolder "3rdParty" like other projects do?

We already have. Bazaar with SVN access. You should read forums and even this thread more carefully.

Mirek
