Subject: SVN plan...
Posted by mirek on Tue, 30 Oct 2007 10:10:07 GMT
View Forum Message <> Reply to Message

Well, just wanted to announce the current idea w.r.t. versioning system(s).

The basic idea is to go SVN way in the future, including TheIDE support.

Anyway, as there are some SVN features I am not very happy about, I will try somewhat more complicated (and expensive) approach.

So in the end, we might create "Uvs3", which would solve these troubles:

- .svn directories in the source tree
- automatic detection of delete/rename -> so that in the end, we have "single click" solution available (makes smart check/commit).
- some sort of local repository to browse at least some revisions (not necessarily all in SVN repository).

Now before being accused about wasting resources on this, I guess solution to these problems is not that complicated and if we are going to integrate SVN with TheIDE, most things uvs3 will have to deal with would have to be implemented for TheIDE/SVN integration as well...

Mirek

Subject: Re: SVN plan...
Posted by mr_ped on Tue, 30 Oct 2007 10:38:48 GMT
View Forum Message <> Reply to Message

My suggestion:
1) integrate SVN as is into TheIDE.
Nothing really fancy, just some basic things per file like update/commit/compare with [working copy/svn]/revert.
Some project level things like add/delete/rename files.
And some informations like current SVN repository/branch for current project/file.

Everything more advanced may be IMHO let on other SVN tools. (SVNtortoise rules)

2) check out other version control systems, I don't have experience with anything else than CVS/SVN (and MS SourceSafe, but that one is sort of joke, not version control system), so I can't help directly, but I think most of the "UVS3" work will be waste of time, because there's probably already some version control system which will work just as you like.

Linux core developers use "git", and ubuntu community does use Bazaar (http://bazaar-vcs.org/) which supports for example distributed repositories (i.e. local ones too).

By Integrating other version control systems into TheIDE you will make just more users happy, so

there's no harm done by this.

What I would not like is the integration of SVN *only* in a "polluted" uvs3 way, that would bring no benefit directly to me.

And there's another thing to consider, where do you want to store UPP sources. The sourceforge has SVN, you should either migrate there completely (that means uvs3 = whatever it will do, in the end it will have to update classic SVN correctly = doesn't sound easy to me, especially if more than 1 developer will work on sourceforge SVN). Or you will end with same situation like now, when there's simply no motivation to fix upp sources extensively, as to propagate the changes to upstream is slow and tedious process.

Subject: Re: SVN plan...
Posted by mdelfede on Tue, 30 Oct 2007 11:13:25 GMT
View Forum Message <> Reply to Message

luzr wrote on Tue, 30 October 2007 11:10Well, just wanted to announce the current idea w.r.t. versioning system(s).

The basic idea is to go SVN way in the future, including TheIDE support.

Quote:
Anyway, as there are some SVN features I am not very happy about, I will try somewhat more complicated (and expensive) approach.

So in the end, we might create "Uvs3", which would solve these troubles:

- .svn directories in the source tree
- automatic detection of delete/rename -> so that in the end, we have "single click" solution available (makes smart check/commit).
- some sort of local repository to browse at least some revisions (not necessarily all in SVN repository).

Well, here just a thought... That depends on *what* you want to do with svn. If it's just a way to distribute daily snapshots to all people, you can keep using uvs2, and I could make some scripts for you that keeps svn on sourceforge up-to-date with your uvs2 repo. OR I could update the repo by myself.
It seems to me that you like much your uvs2... so why change ?

OTOH, if you want to use the repository for a collaborative development.... things change. IMHO svn is not perfect but does a quite good job. I usually prefere the idea of a central repository, but that doesn't block you to have a local read-only copy to make tests. Automate a simple task such adding new files is a matter of 1-liner script. If you want to automate deletions too, that becomes a

bit more complicated, but it's no problem too. For renames, I'd leave it manual.
To resume, svn is rock solid and quite easy to use, even not perfect.
As I told you before, I also tested mercurial, is quite good and modern design, solves many svn/uvs caveats but... It uses local repositories. Nothing bad *if* all developers keep sharing their repos, *very* bad if they don't.
There are also some GUIs for it, and it's not difficult to put a shared repo on sourceforge, neither... it must just be uploaded with ftp (or maybe rsync, I think). So you'd have your local repo and could put an updated copy on sourceforge. IMHO Git and Bazaar work the same way, with local repos, with the disadvantage that they're not so developed as mercurial on windows platform.

Quote:
Now before being accused about wasting resources on this.....

I think you can "waste" your resources as you like... Open Source programming is mostly for fun!

Ciao

Max

Subject: Re: SVN plan...
Posted by tvanriper on Thu, 08 Nov 2007 12:16:27 GMT
View Forum Message <> Reply to Message

Have you looked into Bazaar?

Subject: Re: SVN plan...
Posted by unodgs on Thu, 08 Nov 2007 14:40:49 GMT
View Forum Message <> Reply to Message

Well, it needs python to run.. We don't wont to make our distribution bigger.

Subject: Re: SVN plan...
Posted by tvanriper on Fri, 09 Nov 2007 11:45:09 GMT
View Forum Message <> Reply to Message

Yeah, that is a problem.

Subject: Re: SVN plan...

Posted by mdelfede on Fri, 09 Nov 2007 13:26:50 GMT
View Forum Message <> Reply to Message

If you want to test something "modern", there is mercurial.... same principle as bazaar, I guess, so local repositories. And is quite well ported on linux + windows too, as opposite to GIT.
It has not the automatic version merging... you must solve code merges manually, with some merge program. I think that was a coder's choice; the rest seems quite good.

Ciao

Max

Subject: Re: SVN plan...
Posted by cbpporter on Sat, 10 Nov 2007 16:33:29 GMT
View Forum Message <> Reply to Message

I don't understand what is so wrong with SVN?

Subject: Re: SVN plan...
Posted by mirek on Sun, 11 Nov 2007 22:41:29 GMT
View Forum Message <> Reply to Message

cbpporter wrote on Sat, 10 November 2007 11:33I don't understand what is so wrong with SVN?

Quote:
...which would solve these troubles:

- .svn directories in the source tree
- automatic detection of delete/rename -> so that in the end, we have "single click" solution available (makes smart check/commit).
- some sort of local repository to browse at least some revisions (not necessarily all in SVN repository).

Note that an "aplogy" for uvs3 is that theide/svn integration is often demanded feature and most of the work on uvs3 will be reused there..

Mirek

Subject: Re: SVN plan...
Posted by mr_ped on Sun, 11 Nov 2007 23:25:53 GMT
View Forum Message <> Reply to Message

".svn directories in the source tree"

I'm not sure what's so wrong with these, never really got into my way.

"automatic detection of delete/rename -> so that in the end, we have "single click" solution available (makes smart check/commit)."

a) personal preference, I personally prefer to do this manually as I often have some files in directory which are only local and shouldn't be added to repository.
In the end I can't even imagine "one click update" working for me, as sometimes I start to work on two changes in the same time, and commit them afterwards by hand as two distinctive commits, if those changes are completely independent and in different files.

b) TheIDE can on package management level do this, so the files within .upp will be automatically deleted/added into svn with next commit.

"some sort of local repository to browse at least some revisions (not necessarily all in SVN repository)."

offline usage? I don't know how to solve this one effectively (svn has no direct support for local repositories), there's some way export/import repository so doing a local ("read only") copy is possible, but it would be a standalone copy not connected with main repos and I have no idea about means of syncing to main repos (without another full export/import).

The svn is based around central server, so something like committing to local repos is not possible at all. (unless you take all the commits from local repos and redo them on the central server again = lot of hassle)

Subject: Re: SVN plan...
Posted by mdelfede on Sun, 11 Nov 2007 23:49:24 GMT
View Forum Message <> Reply to Message

mr_ped wrote on Mon, 12 November 2007 00:25".svn directories in the source tree"

I'm not sure what's so wrong with these, never really got into my way.

on Linux they don't appear (hidden files), on windows they may disturb a bit... but I agree, not a real problem

Quote:
"automatic detection of delete/rename -> so that in the end, we have "single click" solution available (makes smart check/commit)."

a) personal preference, I personally prefer to do this manually as I often have some files in directory which are only local and shouldn't be added to repository.
In the end I can't even imagine "one click update" working for me, as sometimes I start to work on two changes in the same time, and commit them afterwards by hand as two distinctive commits, if

those changes are completely independent and in different files.

b) TheIDE can on package management level do this, so the files within .upp will be automatically deleted/added into svn with next commit.

I agree, that should be separate from svn plugin. TheIDE *can* send an additional command to svn to tell him to add or delete files, it's quite easy, but that should be done only for ide managed files. It should not touch files outside packages.

Quote:
"some sort of local repository to browse at least some revisions (not necessarily all in SVN repository)."

offline usage? I don't know how to solve this one effectively (svn has no direct support for local repositories), there's some way export/import repository so doing a local ("read only") copy is possible, but it would be a standalone copy not connected with main repos and I have no idea about means of syncing to main repos (without another full export/import).

The svn is based around central server, so something like committing to local repos is not possible at all. (unless you take all the commits from local repos and redo them on the central server again = lot of hassle)

Well, that's the usual point on local/remote repositories.
I do prefere a central repo too, maybe with a local read-only one kept in sync for regression tests (it's much faster).
Inside theide it would be nice to have the (optional) ability to create and mantain an additional local svn repo, in sync with the remote one. It would also be interesting the possibility to 'go back' to an older revision locking the repository in order to avoid unwanted updates during regression tests and, maybe, the ability to check out an arbitrary revision on a different local path, for tests purposes.

Max

Subject: Re: SVN plan...
Posted by mr_ped on Mon, 12 Nov 2007 08:12:45 GMT
View Forum Message <> Reply to Message

Quote:It would also be interesting the possibility to 'go back' to an older revision locking the repository in order to avoid unwanted updates during regression tests and, maybe, the ability to check out an arbitrary revision on a different local path, for tests purposes.

Why locking out updates, if you work with revision numbers? During tests?

Thinking about local test of different revision...

As the include paths within packages are relative to nests, this sounds a bit cumbersome inside TheIDE right now.
You can't use different package name, that would break include paths in sources.
I wonder what would happen if one would have two packages with same name in two different nests. Probably some havoc.

It looks to me like the least problematic way to go is to "clone" your current nests, disable the original ones, check out older revisions into cloned nests, rebuild all, test.
Possible to do by hand with TheIDE already now, but some additional support to clone/enable/disable nests would be maybe nice.

But I think it should be either as complete as possible to make it almost something like "single" click experience, or there's no point to adding anything and current status will do. Going only half way will not make happy less experienced users, and the more advanced users will save only couple of second or minutes, so I think it would not really improve the current situation.

Subject: Re: SVN plan...
Posted by mdelfede on Mon, 12 Nov 2007 13:14:10 GMT
View Forum Message <> Reply to Message

mr_ped wrote on Mon, 12 November 2007 09:12Quote:It would also be interesting the possibility to 'go back' to an older revision locking the repository in order to avoid unwanted updates during regression tests and, maybe, the ability to check out an arbitrary revision on a different local path, for tests purposes.

Why locking out updates, if you work with revision numbers? During tests?

I'll make an example: suppose you're in revision 50, you want to see what happened to 49... so you request the 49 one.
Now, you go to drink a coffee, you return to PC, forgetting you did request 49 version and start working on your files... you update SVN and you get a 51 version that is a modified 49 one, not 50... Ok, that's solvable, but not nice thing.
That I wanted to say speaking about locking. Here you could :
1- Put IDE in read only mode, if you're looking in a previous fetched revision (a bit drastic way, but it works...)
2- When you save files, IDE should tell you that you're doing it on a past revision
3- Fetch past revision in a separate tree.
The 1 is the fastest, the 2 is half a way between speed and security and the 3d is slow but most secure.

Quote:
Thinking about local test of different revision...

As the include paths within packages are relative to nests, this sounds a bit cumbersome inside TheIDE right now.
You can't use different package name, that would break include paths in sources.
I wonder what would happen if one would have two packages with same name in two different

nests. Probably some havoc.

With theide I usually keep 1 source tree for the stable one in ~/sources/upp, and another one in ~/sources/uppdev with current uvs tree. I use the upp one to compile the uppdev, adding a nest to the uppdev tree in the upp nests. So, I *do* have many packages with same names on different nests, and that makes no problem at all... The important thing is that nests must not overlap. I've got an 'uppsrc' nest with stable ide, and an 'uppdev' nest with devel one pointing to different stuffs. Of course, you can't mix both !

Quote:
It looks to me like the least problematic way to go is to "clone" your current nests, disable the original ones, check out older revisions into cloned nests, rebuild all, test.
Possible to do by hand with TheIDE already now, but some additional support to clone/enable/disable nests would be maybe nice.

That's my 3rd point above.... doew work, but it's slow !
IMHO the best would be fetch past revision in current tree AND provide some locking mechanics to avoid editing of files when an older revision is fetched in.
Just another example : you're working on rev. 50, you find a bug. You fetch past revisions, doing a binary search, and you find that the bug appears between 44 and 45. You do it recompiling each time, but it should be fast enough, as only few files change. So, you find the bug, ask theide to create a diff between buggy and non buggy code, revert to latest version and apply the patch, or correct manually the code.
That could work on separate trees, but would be awfully slow...

Quote:
But I think it should be either as complete as possible to make it almost something like "single" click experience, or there's no point to adding anything and current status will do. Going only half way will not make happy less experienced users, and the more advanced users will save only couple of second or minutes, so I think it would not really improve the current situation.

I agree. IMHO, that should be a single click to "update svn" that takes care of file adding/removing/changes on svn repo, and the same for going back. And it should be limited to files managet from theide, so listed in UPP files.
Having a locally synchronized svn repo could be a plus.

Ciao

Max

Subject: Re: SVN plan...
Posted by mr_ped on Mon, 12 Nov 2007 13:42:04 GMT
View Forum Message <> Reply to Message

mdelfede wrote on Mon, 12 November 2007 14:14
That's my 3rd point above.... doew work, but it's slow !

IMHO the best would be fetch past revision in current tree AND provide some locking mechanics to avoid editing of files when an older revision is fetched in.

The slow, but most safe solution.
With going back in the same directory you will keep the unversioned local files available, which may affect the running of executable.
While I tend to keep local files around, I prefer to have in SVN always fully build-able sources, and this can be tested reliably only by check-outing into empty directory and building from scratch.

mdelfede wrote on Mon, 12 November 2007 14:14
Just another example : you're working on rev. 50, you find a bug. You fetch past revisions, doing a binary search, and you find that the bug appears between 44 and 45. You do it recompiling each time, but it should be fast enough, as only few files change. So, you find the bug, ask theide to create a diff between buggy and non buggy code, revert to latest version and apply the patch, or correct manually the code.
That could work on separate trees, but would be awfully slow...

You think you will compile only files which did change during going to different revision?
So far from my experience with TheIDE the dependency check is extremely unreliable, I often have to do "Rebuilt all" twice a day (as the only package which I can safely assume is compiled correctly already is "Core", and it's too much work to select+clean package other 13 of them by hand) to get correct binary.
If I would find with your method that the bug is in between 44 and 45, I would think twice if I should firstly check source difference (if I can spot the bug quickly), or to rebuild them both from scratch to make sure it's really those two and not some old object file hunting the test results. (especially with U++ and TheIDE)

Another thing is that with fresh cloned nests you can keep several revision on the disk at the same time, making the switching between them and comparing easy even for tools which are not aware of SVN.

It's surely slower, but with modern 2GHz Core2 CPU I can rebuild my current project in 10sec, what is absolutely fantastic for me.
If the rebuild would take 2mins, I would probably like the option of your approach, just to compile modified files and worry about binary is trustworthy after I see strange results.

So I think the speed issue depends a lot on the HW used and project size.

About single button for SVN update...
From Mirek posts I have the feel he uses the same approach even for commits?
While I appreciate simple Update (3 click with TortoiseSVN: right click on the file folder, left click on the "Update" menu, left click on the "OK" button in results window with modified files listed), I do need to write commit log comments and to check out modified files ahead of commit one more time, if everything looks ok. So any simple commit button will not really work for me.

So we have already 3 different views on SVN usage (from Mireks "it's not good for me" trough

yours and my little details and differences) ... This will be a tough thing to do it right.

Subject: Re: SVN plan...
Posted by mdelfede on Mon, 12 Nov 2007 13:57:47 GMT
View Forum Message <> Reply to Message

mr_ped wrote on Mon, 12 November 2007 14:42

So we have already 3 different views on SVN usage (from Mireks "it's not good for me" trough yours and my little details and differences) ... This will be a tough thing to do it right.

Well, 'configuratione pages' are there for something, isn't it ?

Ciao

Max

Subject: Re: SVN plan...
Posted by zsolt on Mon, 12 Nov 2007 22:47:02 GMT
View Forum Message <> Reply to Message

Isn't it possible, to write a general "version control plugin system"?
Something like in eclipse?

A method to get an icon for a particular file in browser, a hook to add extra items to popup-menu of an item in browser, and some hooks on file addition, renaming and deletion.
Of course my list may be incomplete.

With a plugin system, it would be possible to create VCS plugins for simple daily use. For advanced tasks, the VCS's native tools could be used.

Subject: Re: SVN plan...
Posted by cbpporter on Wed, 14 Nov 2007 15:27:04 GMT
View Forum Message <> Reply to Message

This is certainly going to take some time and I need some versioning system right now for U++ sources. This is a little of topic, but can I upload the sources of U++ to some free SVN server? Or is there a way to use Uvs2 or whatever you are using right now. I can host my own server, but if there is complex process of setting up a Uvs repository I would take the easy was and just use a free public SVN.

Subject: Re: SVN plan...
Posted by mdelfede on Wed, 14 Nov 2007 16:04:17 GMT
View Forum Message <> Reply to Message

cbpporter wrote on Wed, 14 November 2007 16:27This is certainly going to take some time and I need some versioning system right now for U++ sources. This is a little of topic, but can I upload the sources of U++ to some free SVN server? Or is there a way to use Uvs2 or whatever you are using right now. I can host my own server, but if there is complex process of setting up a Uvs repository I would take the easy was and just use a free public SVN.

AFAIK to host an UVS server you just need to :

1) Download the full UVS2 repo and put it on a folder on you server
2) Give (anonymous) ftp access to people to that folder

To keep in sync then you must fetch from UVS2 main server just the added version files and 2 index files and put on your server.

Ciao

Max

p.s.: I'd do it by myself but I haven't a free ftp server handy....