Subject: GIT support integration to the IDE
Posted by coolman on Tue, 08 Nov 2016 11:31:36 GMT

View Forum Message <> Reply to Message

Hi.

I try to implement GIT support to the IDE. You can find the actual implementation at https://github.com/CoolmanCZ/mirror.
Currently is supported:
* basic integration with IDE to support manipulation with the files
* GIT configuration
* GIT branch / remote configuration
* fetch, merge, push
* commit
* stash / stash apply
* GIT repository history
* GIT file history

I would like to ask you to help me with the testing, also suggestions are welcome. Please, take into account that the current implementation is a prototype. So It can be changed/enhanced by your requests.
For your requirements, please use the issue tracker at the link mentioned above.

Thanks, Radek

Subject: Re: GIT support integration to the IDE
Posted by mr_ped on Tue, 08 Nov 2016 22:24:13 GMT

View Forum Message <> Reply to Message

I'm quite content with using git-cola in second windows, but one thing would be very nice for me.

"Annotate" (called "blame" few years back :)) on current tab, so every line would have line number on left with color-per-commit + commint author, and on mouse hover the commit message would be displayed (I like the look of this feature in Android Studio (IntelliJ based)).

Subject: Re: GIT support integration to the IDE
Posted by deep on Wed, 09 Nov 2016 14:44:52 GMT

View Forum Message <> Reply to Message

Hi,

Tested TheIDE with git.
Working good on first use.

Downloaded and compiled from your sources.

Then added .git folder of git clone from "https://github.com/ultimatepp/mirror"
Checked with https://github.com/ultimatepp/mirror

Also working with internally hosted gitlab server.

Nice work. Will use it. I was looking for it.

"git blame" may be useful

Subject: Re: GIT support integration to the IDE
Posted by deep on Sat, 19 Nov 2016 14:09:05 GMT
View Forum Message <> Reply to Message

Hi Radek,

Can you give this in diff format.

Will be easy to add to the current HEAD.

Subject: Re: GIT support integration to the IDE
Posted by coolman on Sat, 19 Nov 2016 14:33:02 GMT
View Forum Message <> Reply to Message

deep wrote on Sat, 19 November 2016 15:09
Can you give this in diff format.

Hi,

I'm sorry, but I don't understand your request. Can you be more specific?

Radek

Subject: Re: GIT support integration to the IDE
Posted by deep on Sat, 19 Nov 2016 15:16:32 GMT
View Forum Message <> Reply to Message

Hi,

The present Trunk in main repo is different than what you have used.

To use your modifications with current HEAD in git will require the patch for your changes.

You used some branch of IDE and then created your additions for git integration.

Can you create patch with ref to your original ide branch.

I would like to use patch to current HEAD.

Subject: Re: GIT support integration to the IDE
Posted by coolman on Sun, 20 Nov 2016 19:21:33 GMT
View Forum Message <> Reply to Message

> deep wrote on Sat, 19 November 2016 16:16Hi,
>
> The present Trunk in main repo is different than what you have used.
>
> To use your modifications with current HEAD in git will require the patch for your changes.
>
> You used some branch of IDE and then created your additions for git integration.
>
> Can you create patch with ref to your original ide branch.
>
> I would like to use patch to current HEAD.

Hi,

Thanks for the explanation. But you can create the diff yourself.
1. Clone the GIT repository git clone https://github.com/CoolmanCZ/mirror upp_git
2. Add remote upstream to the trunk repository cd upp_git
git remote add upstream https://github.com/ultimatepp/mirror
3. Create the diff git diff upstream/master..HEAD

When you are using the IDE from the U++ GIT repository and you already have the upstream configured, you don't need the patch. You can merge the changes from the upstream to the U++ GIT repository using integrated merge function.

Best regards, Radek

Edit: I am trying to keep the U++ GIT repository up to date with the latest trunk version of the IDE.

Subject: Re: GIT support integration to the IDE
Posted by deep on Mon, 21 Nov 2016 04:43:32 GMT
View Forum Message <> Reply to Message

Hi Radek,

coolman wrote on Mon, 21 November 2016 00:51Edit: I am trying to keep the U++ GIT repository up to date with the latest trunk version of the IDE.

This will be great.

Subject: Re: GIT support integration to the IDE
Posted by coolman on Wed, 24 Jan 2018 16:52:32 GMT
View Forum Message <> Reply to Message

Hi all,

After a long pause I've completed the basic functionality of the U++ GIT support. I'd like to ask you for the testing. If you find a bug create an issue at the GitHub, please. Do the same if you have any suggestions / requests.

Thank you, Radek

Subject: Re: GIT support integration to the IDE
Posted by Klugier on Wed, 24 Jan 2018 20:13:38 GMT
View Forum Message <> Reply to Message

Hello Radek,

I see you integrated your U++ branch with travis CI. How do you achieve that? I am strongly interested in doing the same for U++ mirror. Can you tell me more about this?

Sincerely,
Klugier

Subject: Re: GIT support integration to the IDE
Posted by coolman on Thu, 25 Jan 2018 06:40:44 GMT
View Forum Message <> Reply to Message

Hi,

In the travis.yml you can see, what have to be installed on the Travis build machine. I've been using the upp_cmake script as git submodule. And the script ./generate_cmake_ide.sh is used to generate CMakeLists.txt files and build the IDE. I've got the scripts to generate and build autotest, examples, references, tutorial and upptst, but these are not in the git repository yet. I can put them to the repository if you are interested in.

I've been thinking about the automatic cross building of the Windows binary of the IDE every week to pull the tarbal to the GitHub using the Travis, but right now it doesn't make the sense. I don't know if anybody else using this fork. :)

BR, Radek

Subject: Re: GIT support integration to the IDE
Posted by mirek on Thu, 25 Jan 2018 10:12:24 GMT
View Forum Message <> Reply to Message

coolman wrote on Wed, 24 January 2018 17:52Hi all,

After a long pause I've completed the basic functionality of the U++ GIT support. I'd like to ask you for the testing. If you find a bug create an issue at the GitHub, please. Do the same if you have any suggestions / requests.

Thank you, Radek

BTW, U++ Synchronize now supports GIT as well (since 2017.2)...

The general idea is perhaps somewhat different from what you would expect from "real GIT support" - the idea, just as with SVN synchronize, is to automate repetitive tasks (commit / push / pull).

Subject: Re: GIT support integration to the IDE
Posted by coolman on Thu, 25 Jan 2018 11:11:31 GMT
View Forum Message <> Reply to Message

[quote title=mirek wrote on Thu, 25 January 2018 11:12]coolman wrote on Wed, 24 January 2018 17:52Hi all,

BTW, U++ Synchronize now supports GIT as well (since 2017.2)...

The general idea is perhaps somewhat different from what you would expect from "real GIT support" - the idea, just as with SVN synchronize, is to automate repetitive tasks (commit / push / pull).

Hi,

Yes, I know about this feature. It can be used for synchronization.

But I needed something, e.g. for basic resolving the conflict during the merge, ... To minimize the situation with switching to the CMD to resolve the simple GIT problems.

BR, Radek

Subject: Re: GIT support integration to the IDE
Posted by coolman on Thu, 07 Jun 2018 11:25:26 GMT
View Forum Message <> Reply to Message

Hi,

I would like to invite you to try GIT menu implementation (develop branch) to the IDE.

BR, Radek

The current implementation includes the following options for manipulating a GIT repository

GIT init repository
Can be used to initialize the GIT repository for selected package if the package is not a part of already initialized GIT repository. When initialization is successfully done GIT config is invoked.

GIT config
It allows you to change the configuration parameters of individual GIT repositories as well as global GIT settings.
(all available GIT repositories are shown)

GIT history
This item displays the GIT commit history of the currently selected branch. A part of the view is a detailed listing of the currently selected commit or a summary of lines and files changed in this commit.
The right click menu allows you to create a new branch, insert/delete/show tags or create a diff/GIT patch of selected commit.
(all available GIT repositories are shown)

GIT update status
Compares and displays the status of local and remote branches within single GIT repositories.

GIT status
Shows the GIT status of the repository that contains the currently edited file.

GIT fetch
Download objects and refs from all remote repositories. Fetches branches and/or tags (collectively, "refs") along with the objects necessary to complete their histories. Remote-tracking branches are updated.
(git fetch is invoked on all available repositories)

GIT merge
Allows you to incorporate changes from one branch (even remote branch) into the current branch. You can see all changes between selected branches before the merge.
(all available GIT repositories are shown)

GIT stash
Stashing takes the dirty state of your working directory - that is, your modified tracked files and staged changes - and saves it on a stack of unfinished changes that you can reapply at any time.

GIT stash apply
You can select and apply the saved changes you just stashed previously.

GIT commit
Stores the current contents of the index in a new commit along with a log message from the user describing the changes. Before the commit you can see all changes made in the repository and:
- discard changes for a particular file
- basic manipulation with unmerged changes
- add untracked file to the repository
- delete file from the repository
- add file to the .gitignore file
- delete file

note: left doubleclick at the line in the right side of the diff output opens the appropriate file in the editor.
(all available GIT repositories are shown)

GIT push
Updates remote refs using local refs, while sending objects necessary to complete the given refs.
(git push is invoked on all available repositories)

GIT branch
Allows you to add / modify / delete /checkout a local branch. There is also the option to add / modify / delete a remote repository, including pairing a local branch with a remote branch in a remote repository.
(all available GIT repositories are shown)

GIT patch apply/abort
Reads the supplied diff output (i.e. "a patch") and applies it to files. When running from a subdirectory in a repository, patched paths outside the directory are ignored. This command applies the patch and create commits from patches generated by git-format-patch and/or received by email.
If the patch apply has failed, you can abort this patch using this menu option.

Show repository history of the file...
Shows GIT history of the edited file (diff output and detailed description of the commit)

Show repository history of the file lines...
Shows what revision and which author has later modified each row of the edited file.