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.