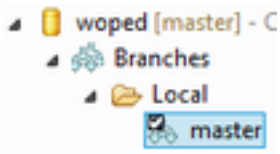
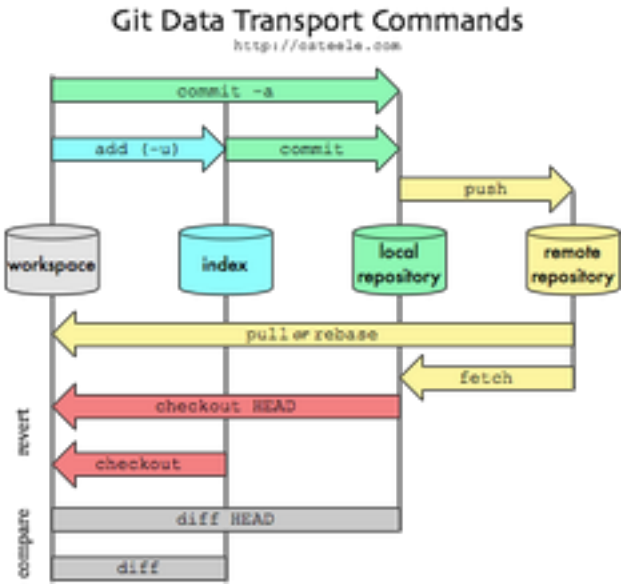
FAQ

Q: Where/in which branch do I create new code/modify existing code?
A: We strongly recommend only modifying the source code within the local master branch: Git Repository View → Woped repository → Branches → Local → master → Checkout. Afterwards, you should see a little check mark beside the branch's name.



Q: How do I commit new changes?
A: Committing changes takes multiple steps:

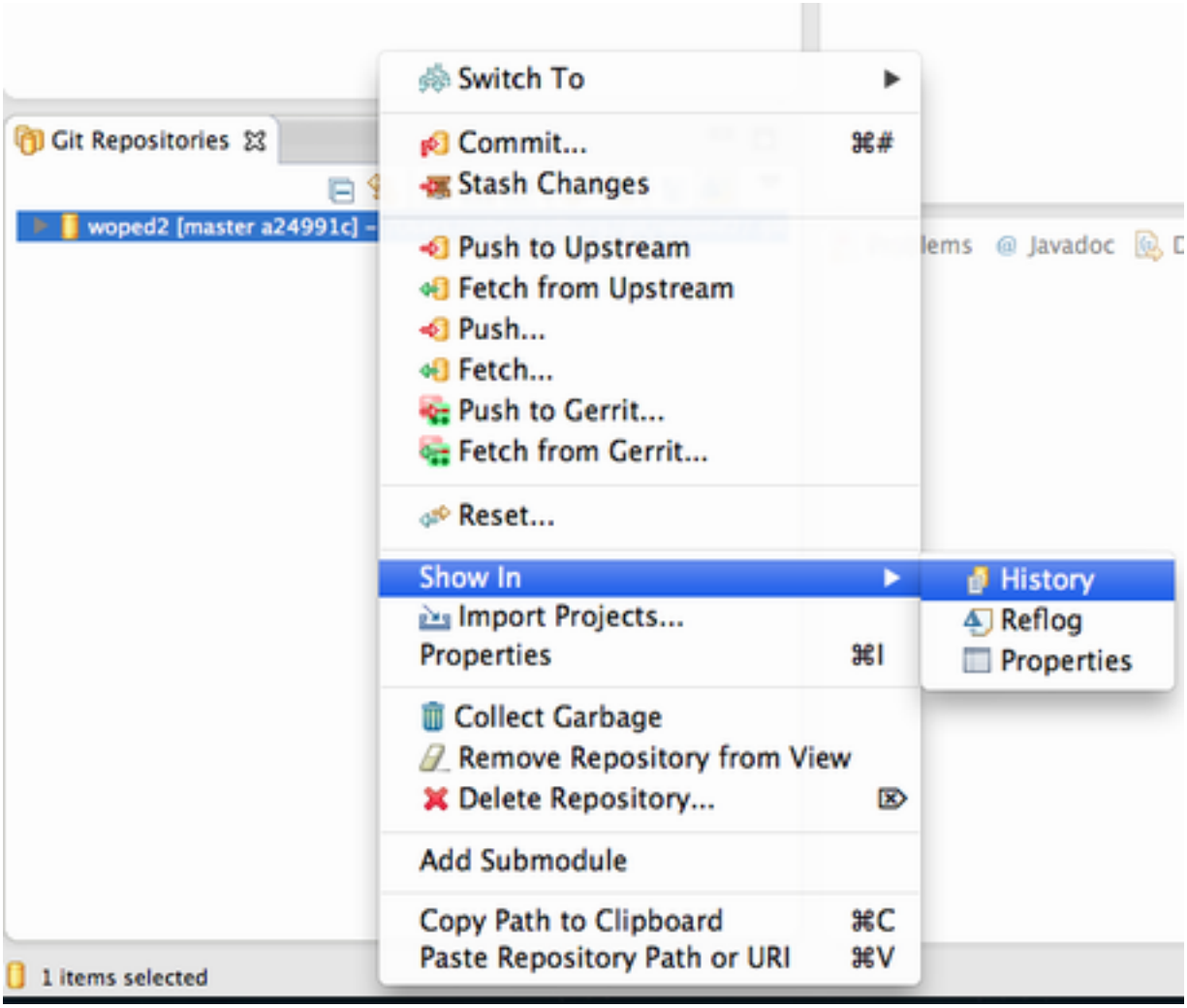
1. Commit
   Commits the selected changes from your local workspace to your local git repository. When your are using EGit's 'commit' command. 'commit -a' is assumed automatically.
2. Push
   Pushes the committed changes from the local git repository to the remote repository, which is provided by your chosen source code hosting service, i. e. Sourceforge, GitHub, etc.



Q: Where can I see the latest changes on the project?

A: In the project history

1. select the git repository you would like to see the history of
2. right click on it and go to ‘Show in...’-
   >’History’



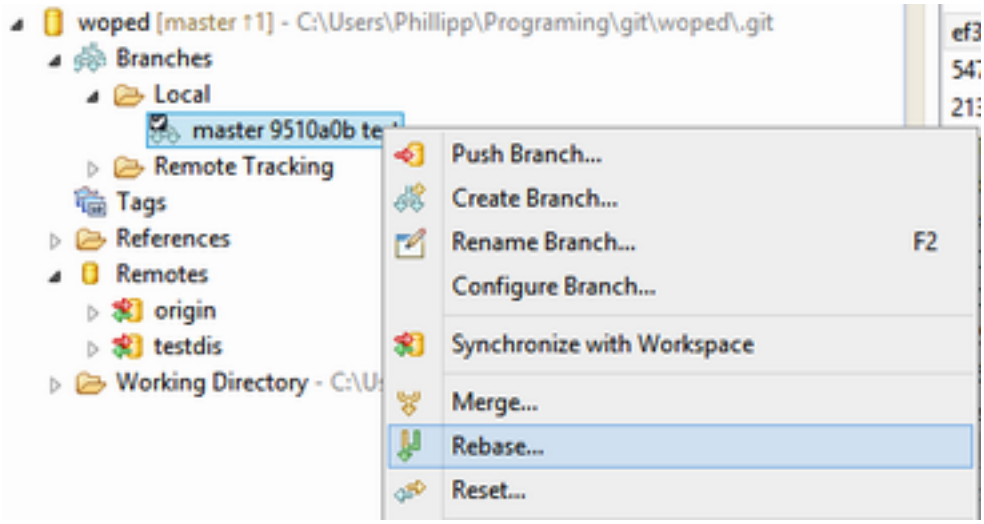
3. In the now opening history tab you see the projects’ history.

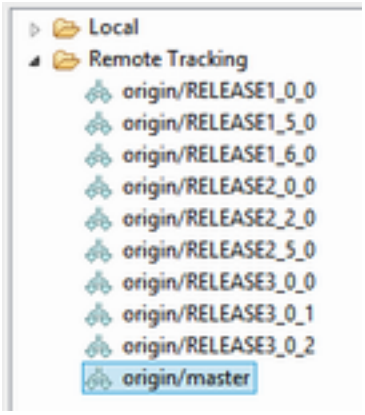
Q: How do I delete file/s?

A: You simply delete the desired file/s in your workspace and commit/push it to the git repositories.

Q: How do I get the changes from other developers?
A: In order to update your local repo with changes from other developers, you have to fetch the current HEAD from the remote repository (regularly the origin/master branch) and merge/rebase them with your local repository. Alternatively you can use the 'pull' command.

Q: What is the difference between fetch and pull?
A: Fetch simply gets the latest version of the git repository from the remote repository (but does not overwrite anything in your workspace or local repository). In order to bring the changes to your local repo, you have to merge or rebase your local master branch with the remote's master branch.



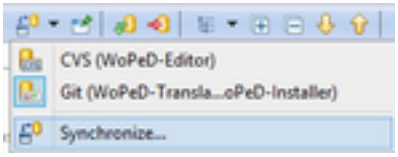


'Pull' does the same as 'fetch', but also merges automatically the changes with your local repository at the same time, i. e. pull = fetch + merge.
Q: How can I see recently pushed changes by other developers?
A: There is more than one way to achieve this goal:

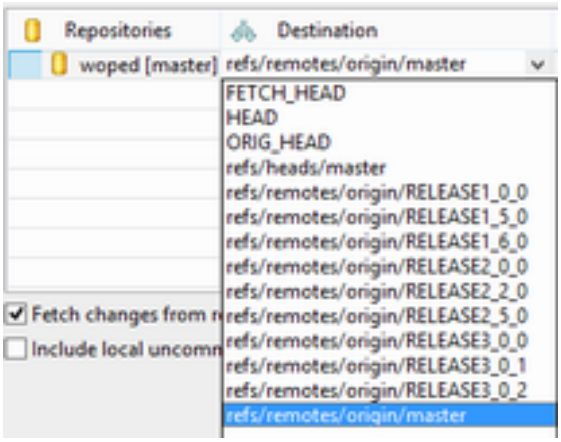
1. Compare a certain file with the remote's current HEAD branch:

Select a file > Compare With > Branch, Tag or Reference > Remote Tracking > **origin/master**

2. Compare the whole workspace with the remote's current HEAD branch:
   Enter the 'Team Synchronizing' view > press 'Synchronize'



   Choose the remote's master branch



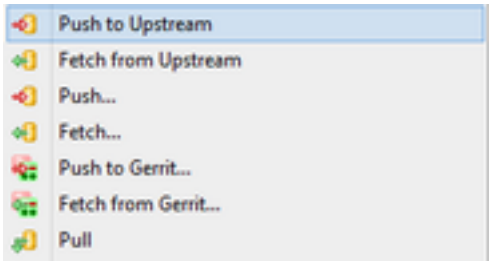
Q: How can I reset my HEAD or my workspace?
A: Show in your 'history' view and select the commit upon you wish to reset your HEAD and/or your workspace. When you click 'Reset' you have 3 options to choose (especially soft and hard resets are useful):

- **soft** - the HEAD points now to the new commit, the index and the working tree are unchanged
- **mixed** - the HEAD points now to the new commit, the index is updated, the working tree is unchanged
- **hard** - the HEAD points now to the new commit, the index and the working tree are updated

Q: What is the best workflow to submit code?

A: We recommend a fetch - review - merge/rebase - commit/push workflow

1. Modify code in your workspace
2. Fetch the current HEAD of your remote repository (regularly the origin/master branch).
3. If there are **no** changes to fetch, it means you have already the latest changes from the remote included in your local repository.
4. If there are changes to fetch, you will see an arrow (pointing down) beside your local repository. The number next to the arrow indicates how many commits your local repository is behind the remote repository.
5. Update your local repository by doing a merge or rebase with the remote repository origin/master branch. Doing so will bring the latest changes to your local repository such that it is up to date again.
   Further information concerning difference between merge and rebase:
   http://www.fiveminutes.eu/git-merge-and-rebase-the-simple-explanation/
6. Now it is time to bring the changes from your workspace to the local repository by simply committing the changes.
7. Last but not least, you have to push the changes from your local repository to the remote repository.
   By using 'Push to Upstream'



Annotation:
Please that 5. and 6. might also be combined by using 'Commit and push'.

Q: How can I adopt changes from the remote repository in my workspace by using the 'Team Synchronizing' view?

A: You can either overwrite or merge single changes within the 'Team Synchronizing' view.

1. Overwrite will overwrite the selected file in your workspace with the file from the remote repository.
2. Merge will try to incorporate changes into your file, i. e. it will assemble one file, which integrates changes from both files (file from your workspace and file from the remote repository).

Important:

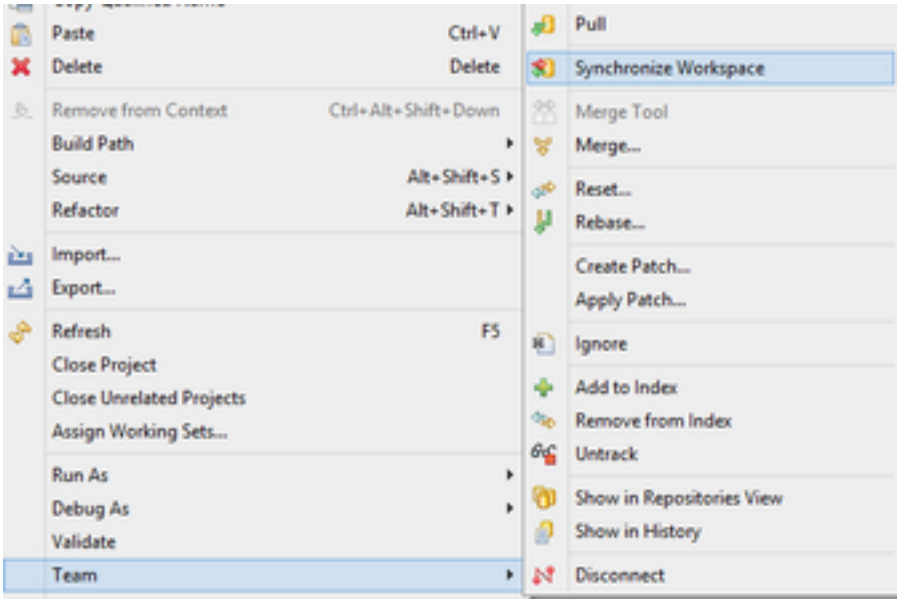
You will not see any changes in the current 'Team Synchronizing' view, since you have just modified files in your **workspace**. Because of that, the changes are currently only in your workspace and **not in your local WoPeD** repository. To bring the changes to your local repository, you have to commit the changes.

Q: What is an alternative workflow to submit code w/o using automated merge/rebase?

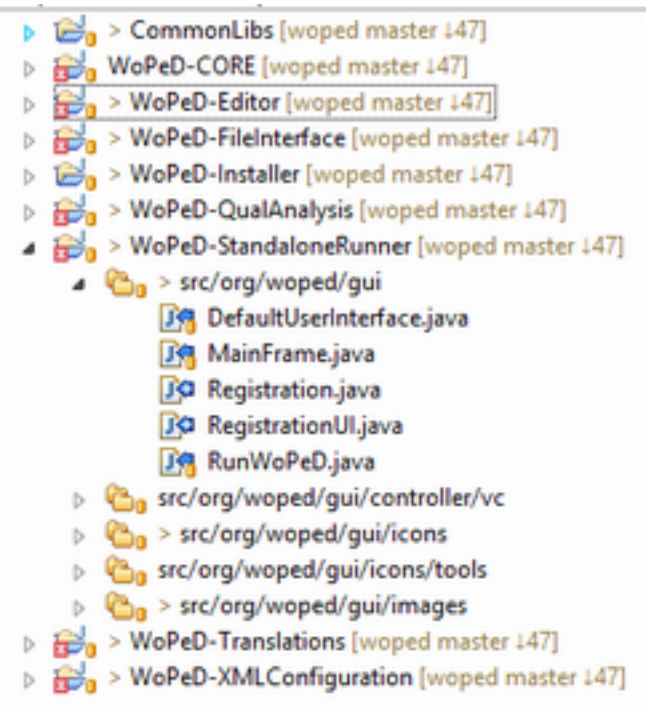
A: We do recommend using the automated workflow, described before. Do the following workflow only if you are aware of what you are doing.

The following might especially important if you are facing merge/rebase errors during the automated workflow.

1. Modify code in your workspace
2. Synchronize your workspace with the remote's HEAD branch (regularly the master branch) → you will fetch the current HEAD files such that you can compare them with the files in your workspace

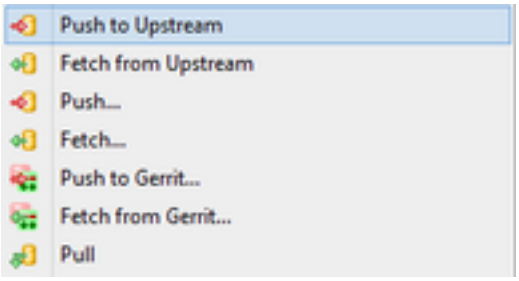


3. In the 'Team Synchronizing' view you are able to see the differences between the files in the workspace and the latest files from the remote repository.



4. Manage the changes between the files either automatically by using merge or overwrite (as described some questions before) or manually by editing the files on your own.

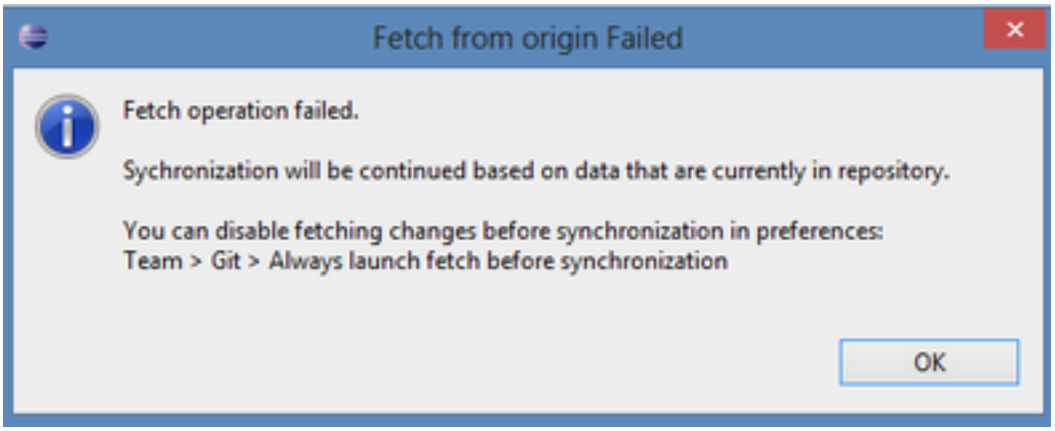
   When you have resolved the differences between the files, you now have all the updated files in your **workspace**.
5. Now you have to update your local repository with the latest version from the remote repository in order to include the latest commits.
6. Afterwards you have to bring the previously edited files from your workspace to the git repositories. Therefore you have to commit the changes to the local repository.
7. Last but not least, you have to push the changes from your local repository to the remote repository.
   By using ‘Push to Upstream’



   Annotation:
   Please that 5. and 6. might also be combined by using ‘Commit and push’.

Troubleshooting

Issue: Fetch failed



Solution: Most likely issue is your network connection.

Issue: Master branch: push non-fast forward
Solution: If you want to push your master branch from your local repository to the remote repository, but somebody else updated the remote's master branch meanwhile you are not allowed to push your master branch anymore, because you do not have the latest commits in your branch's history.
Therefore you have to bring the latest commits from the remote to your local repository. Fetch the latest version from the remote repository and rebase/merge your local repository with the remote repository → up to date. If there are any merging conflicts, you have to resolve them manually by comparing and modifying the concerned file/s.