





# Example of Merge Conflict in Git

Graph	Description
	revision one (in master)

Graph	Description
	revision one (in master)

```
Main.java ●
1  public class Main
2  {
3      //this is the chronological revision 1 in master
4
5      public static void main(String[] args)
6      {
7          System.out.println("Hello world");
8      }
9
10 }
11
```

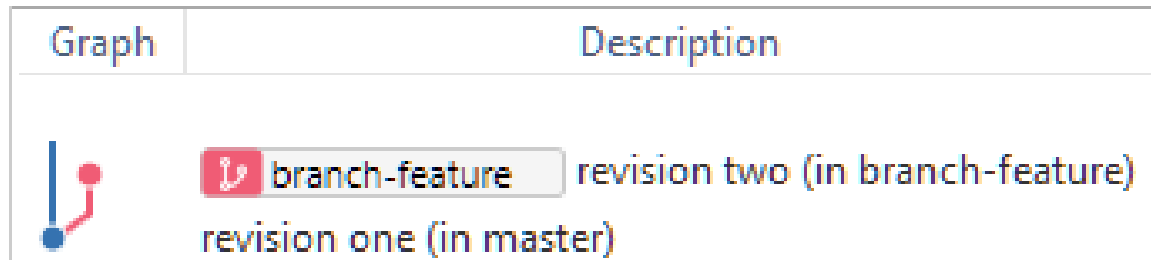
Graph	Description
	<div> <b>branch-feature</b> revision two (in branch-feature)</div> <div>revision one (in master)</div>

```
Main.java •
1  public class Main
2  {
3      //this is the chronological revision 1 in master
4
5      public static void main(String[] args)
6      {
7          System.out.println("Hello world");
8      }
9
10 }
11
```

```

Main.java x
1  public class Main
2  {
3      //this is the chronological revision 2 in branch-feature
4
5      public static void main(String[] args)
6      {
7          System.out.println("Hello world");
8      }
9
10     public void methodInBranch()
11     {
12         System.out.println("Method in branch");
13     }
14 }
15

```



```

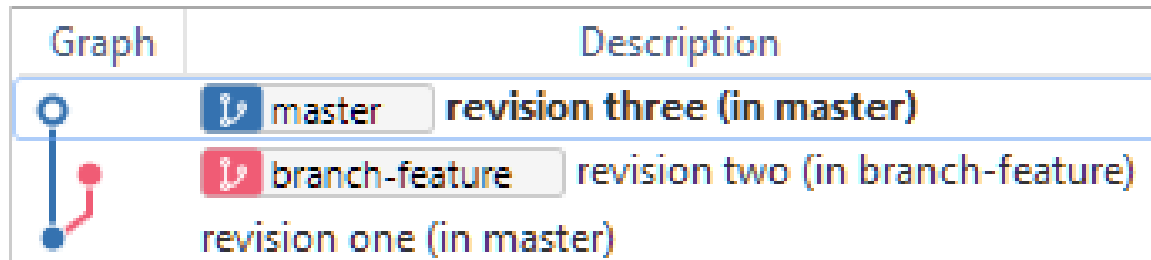
Main.java •
1  public class Main
2  {
3      //this is the chronological revision 1 in master
4
5      public static void main(String[] args)
6      {
7          System.out.println("Hello world");
8      }
9
10 }
11

```

```

Main.java x
1  public class Main
2  {
3      //this is the chronological revision 2 in branch-feature
4
5      public static void main(String[] args)
6      {
7          System.out.println("Hello world");
8      }
9
10     public void methodInBranch()
11     {
12         System.out.println("Method in branch");
13     }
14 }
15

```



```

Main.java •
1  public class Main
2  {
3      //this is the chronological revision 1 in master
4
5      public static void main(String[] args)
6      {
7          System.out.println("Hello world");
8      }
9
10 }
11

```

```

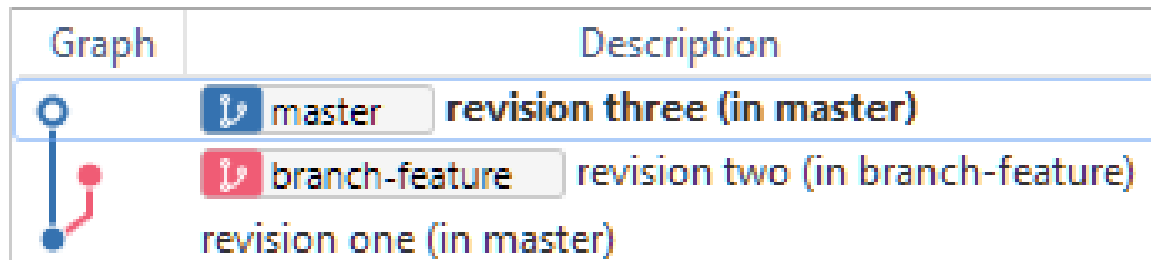
Main.java •
1  public class Main
2  {
3      //this is the chronological revision 3 in master
4
5      public static void main(String[] args)
6      {
7          System.out.println("Hello world");
8      }
9
10     public void methodInMaster()
11     {
12         System.out.println("Method in master");
13     }
14 }
15

```

```

Main.java x
1  public class Main
2  {
3      //this is the chronological revision 2 in branch-feature
4
5      public static void main(String[] args)
6      {
7          System.out.println("Hello world");
8      }
9
10     public void methodInBranch()
11     {
12         System.out.println("Method in branch");
13     }
14 }
15

```






```

Main.java •
1  public class Main
2  {
3      //this is the chronological revision 1 in master
4
5      public static void main(String[] args)
6      {
7          System.out.println("Hello world");
8      }
9
10 }
11




```

# Resolution of Merge Conflict with 2-way-merge

Graph	Description
	<div> master</div> revision three (in master)
	<div> branch-feature</div> revision two (in branch-feature)
	revision one (in master)



# Resolution of Merge Conflict with 2-way-merge

Graph	Description
	 master <b>revision three (in master)</b>
	 branch-feature   revision two (in branch-feature)
	revision one (in master)

In master, run:  
`git merge branch-feature`

Changes are visualized with:




Visual Studio Code

<https://github.com/Microsoft/vscode>

```
Main.java x
1 public class Main
2 {
3     Accept Current Change | Accept Incoming Change | Accept Both Changes | Compare Changes
4     <<<<<< HEAD (Current Change)
5         //this is the chronological revision 3 in master
6     =====
7     //this is the chronological revision 2 in branch-feature
8     >>>>>> branch-feature (Incoming Change)
9
10    public static void main(String[] args)
11    {
12        System.out.println("Hello world");
13    }
14
15    Accept Current Change | Accept Incoming Change | Accept Both Changes | Compare Changes
16    <<<<<< HEAD (Current Change)
17    public void methodInMaster()
18    {
19        System.out.println("Method in master");
20    }
21    =====
22    public void methodInBranch()
23    {
24        System.out.println("Method in branch");
25    }
26    >>>>>> branch-feature (Incoming Change)
27    }
28 }
```

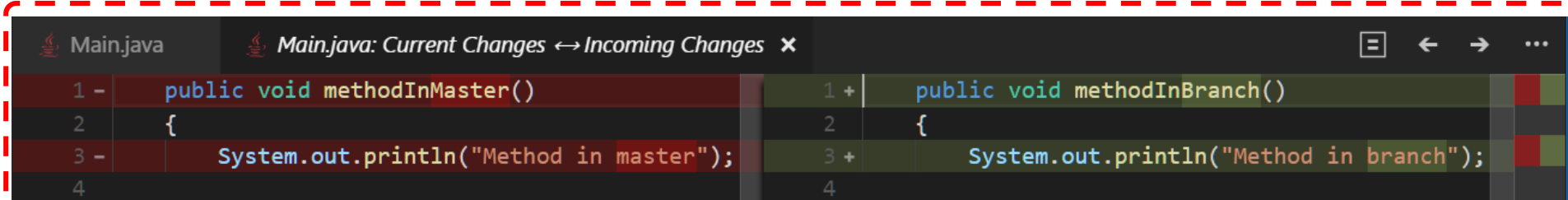
```
Main.java x
1 public class Main
2 {
3     Accept Current Change | Accept Incoming Change | Accept Both Changes | Compare Changes
4     <<<<<< HEAD (Current Change)
5         //this is the chronological revision 3 in master
6     =====
7         //this is the chronological revision 2 in branch-feature
8     >>>>>> branch-feature (Incoming Change)
9
10    public static void main(String[] args)
11    {
12        System.out.println("Hello world");
13    }
14
15    Accept Current Change | Accept Incoming Change | Accept Both Changes | Compare Changes
16    <<<<<< HEAD (Current Change)
17    public void methodInMaster()
18    {
19        System.out.println("Method in master");
20    }
21    =====
22    public void methodInBranch()
23    {
24        System.out.println("Method in branch");
25    }
26    >>>>>> branch-feature (Incoming Change)
27    }
28 }
```

Changes are visualized with:

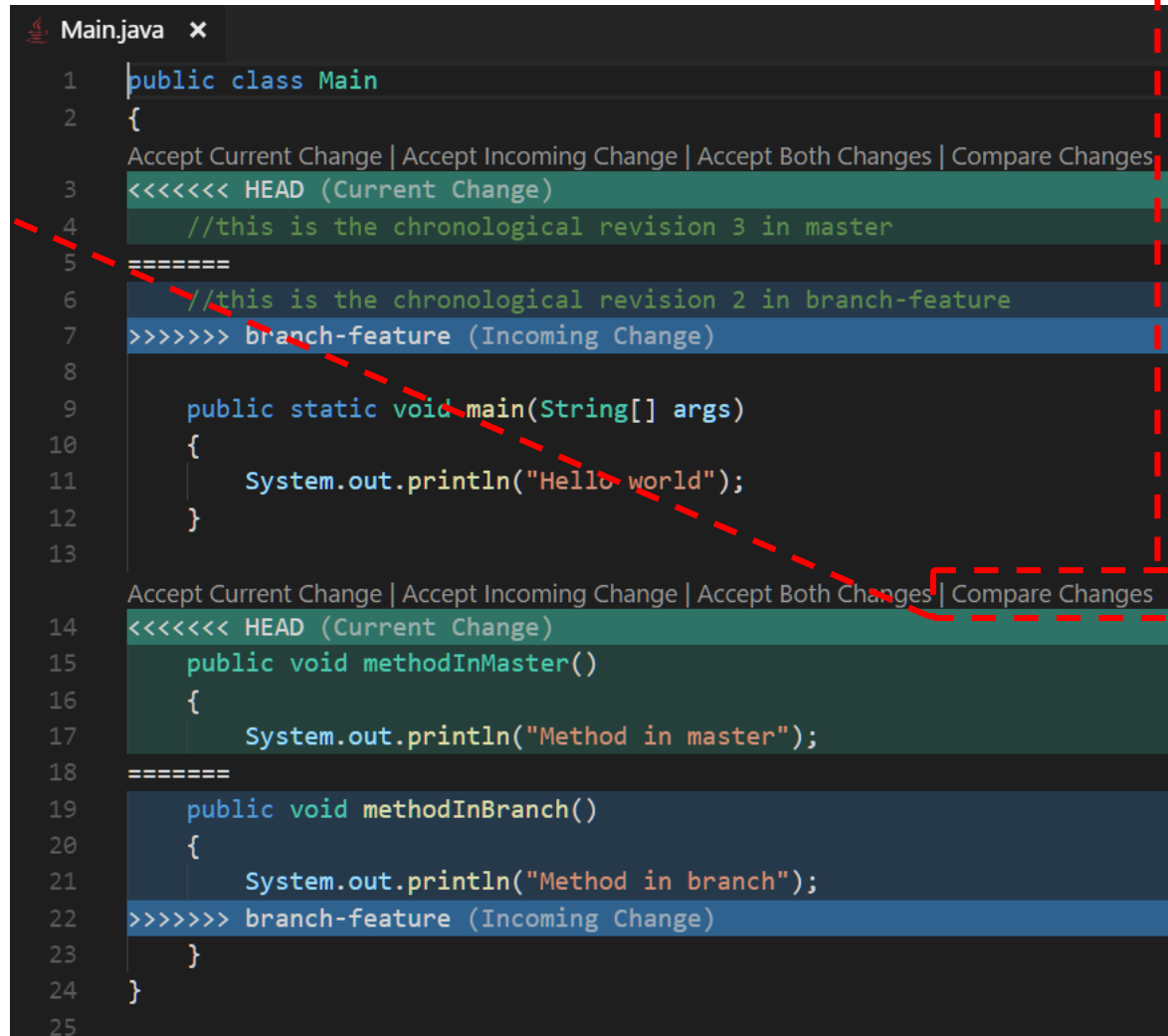


Visual Studio Code

<https://github.com/Microsoft/vscode>

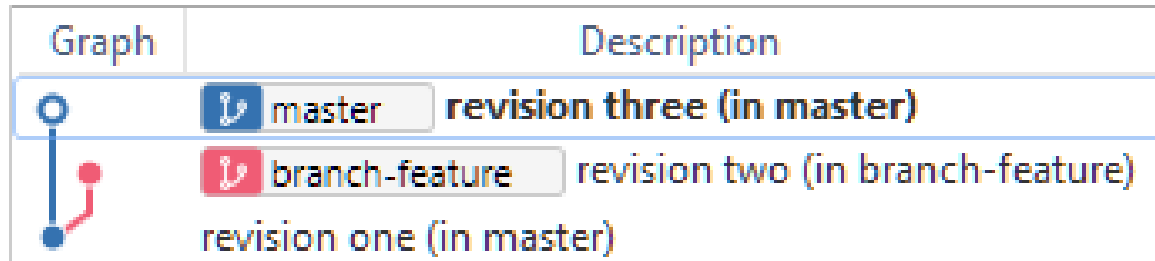


Main.java		Main.java: Current Changes ↔ Incoming Changes	
1 -	public void methodInMaster()	1 +	public void methodInBranch()
2	{	2	{
3 -	System.out.println("Method in master");	3 +	System.out.println("Method in branch");
4		4	



```
1 public class Main
2 {
3     Accept Current Change | Accept Incoming Change | Accept Both Changes | Compare Changes
4     <<<<<< HEAD (Current Change)
5         //this is the chronological revision 3 in master
6     =====
7     //this is the chronological revision 2 in branch-feature
8     >>>>>> branch-feature (Incoming Change)
9
10    public static void main(String[] args)
11    {
12        System.out.println("Hello world");
13    }
14
15    Accept Current Change | Accept Incoming Change | Accept Both Changes | Compare Changes
16    <<<<<< HEAD (Current Change)
17    public void methodInMaster()
18    {
19        System.out.println("Method in master");
20    }
21    =====
22    public void methodInBranch()
23    {
24        System.out.println("Method in branch");
25    }
26    >>>>>> branch-feature (Incoming Change)
27 }
```

# Resolution of Merge Conflict with 3-way-merge






In master, run:

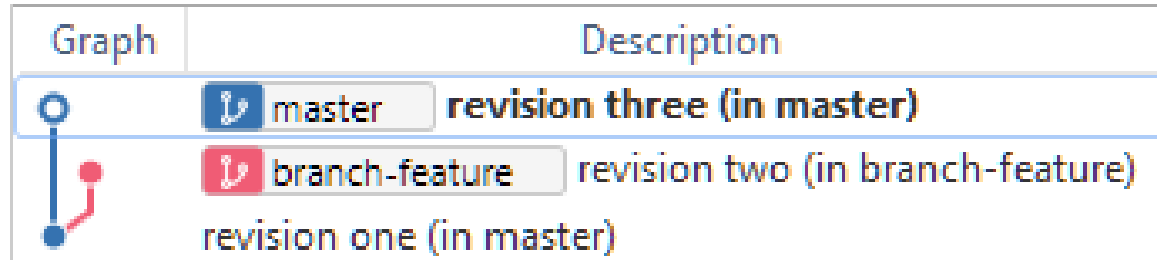
```
git merge branch-feature  
git mergetool
```

- Note: The `git mergetool` was configured to be Perforce P4Merge (from the default vimdiff)
  - <https://www.perforce.com/products/helix-core-apps/merge-diff-tool-p4merge>

# 3-way merge (Perforce P4Merge)

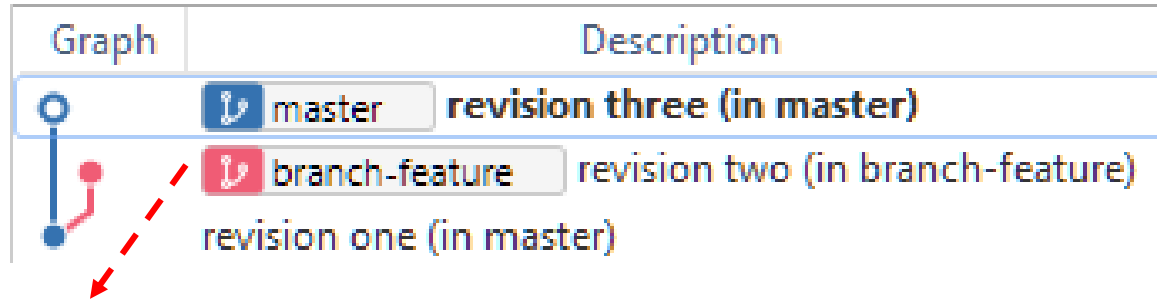
Graph	Description
	 master <b>revision three (in master)</b>
	 branch-feature    revision two (in branch-feature)
	revision one (in master)

# 3-way merge (Perforce P4Merge)



```
./Main_REMOTE_7736.java  
  
public class Main  
{  
    //this is the chronological revision 2 in branch-  
  
    public static void main(String[] args)  
    {  
        System.out.println("Hello world");  
    }  
  
    public void methodInBranch()  
    {  
        System.out.println("Method in branch");  
    }  
}
```

# 3-way merge (Perforce P4Merge)



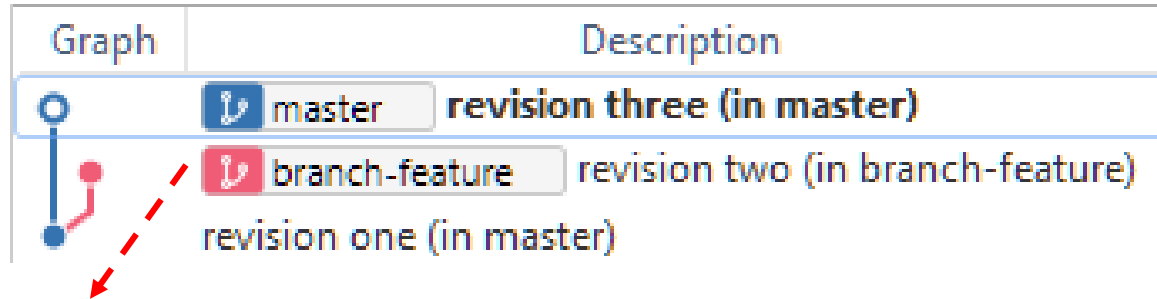
```
./Main_REMOTE_7736.java
public class Main
{
    //this is the chronological revision 2 in branch-

    public static void main(String[] args)
    {
        System.out.println("Hello world");
    }

    public void methodInBranch()
    {
        System.out.println("Method in branch");
    }
}
```



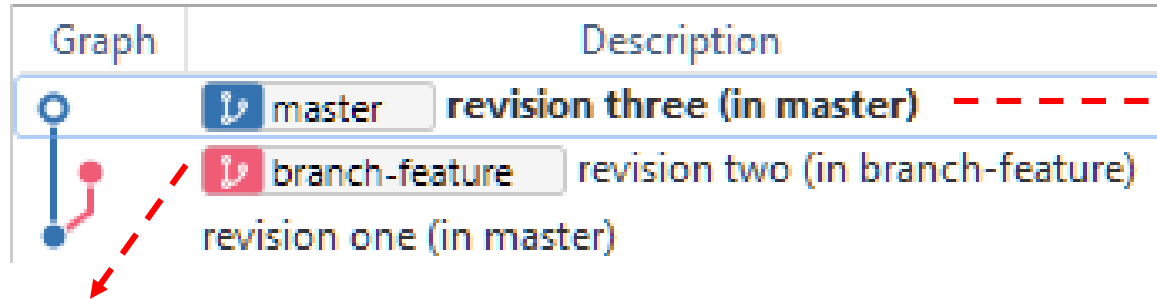
# 3-way merge (Perforce P4Merge)



```
./Main_REMOTE_7736.java
public class Main
{
    //this is the chronological revision 2 in branch-
    public static void main(String[] args)
    {
        System.out.println("Hello world");
    }
    public void methodInBranch()
    {
        System.out.println("Method in branch");
    }
}
```

```
./Main_LOCAL_7736.java
public class Main
{
    //this is the chronological revision 3 in master
    public static void main(String[] args)
    {
        System.out.println("Hello world");
    }
    public void methodInMaster()
    {
        System.out.println("Method in master");
    }
}
```

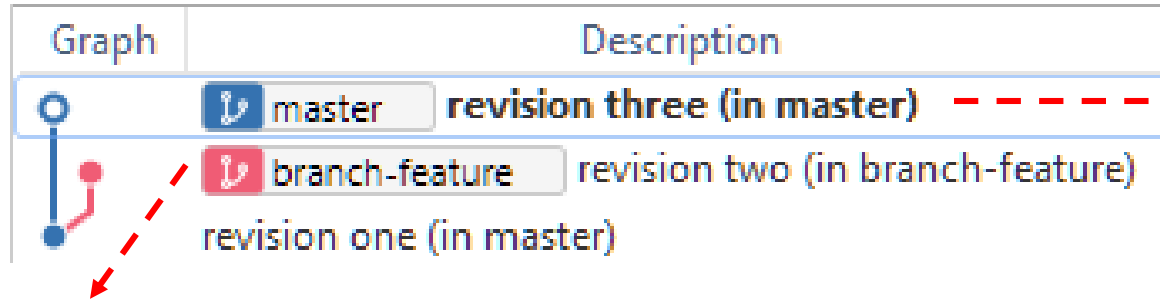
# 3-way merge (Perforce P4Merge)



```
./Main_REMOTE_7736.java  
  
public class Main  
{  
    //this is the chronological revision 2 in branch-  
  
    public static void main(String[] args)  
    {  
        System.out.println("Hello world");  
    }  
  
    public void methodInBranch()  
    {  
        System.out.println("Method in branch");  
    }  
}
```

```
./Main_LOCAL_7736.java  
  
public class Main  
{  
    //this is the chronological revision 3 in master  
  
    public static void main(String[] args)  
    {  
        System.out.println("Hello world");  
    }  
  
    public void methodInMaster()  
    {  
        System.out.println("Method in master");  
    }  
}
```

# 3-way merge (Perforce P4Merge)

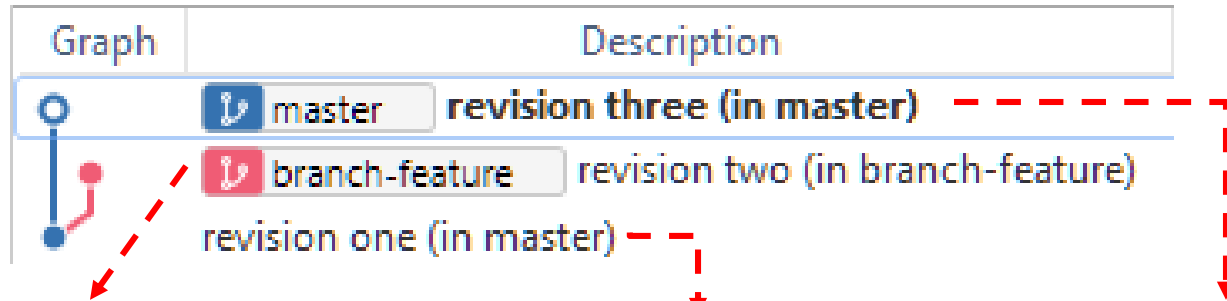


```
./Main_REMOTE_7736.java  
  
public class Main  
{  
    //this is the chronological revision 2 in branch-  
  
    public static void main(String[] args)  
    {  
        System.out.println("Hello world");  
    }  
  
    public void methodInBranch()  
    {  
        System.out.println("Method in branch");  
    }  
}
```

```
./Main_LOCAL_7736.java  
  
public class Main  
{  
    //this is the chronological revision 3 in master  
  
    public static void main(String[] args)  
    {  
        System.out.println("Hello world");  
    }  
  
    public void methodInMaster()  
    {  
        System.out.println("Method in master");  
    }  
}
```

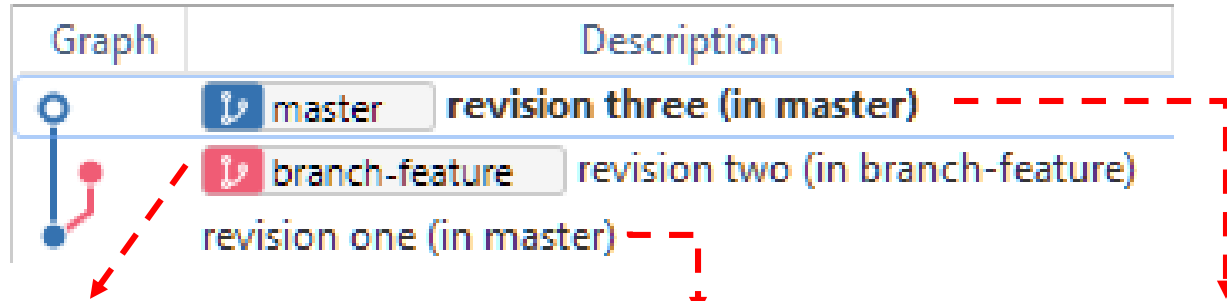
Shows the latest changes in both branches  
(same as a 2-way merge)

# 3-way merge (Perforce P4Merge)



./Main_REMOTE_7736.java	./Main_BASE_7736.java	./Main_LOCAL_7736.java
<pre>public class Main {     //this is the chronological revision 2 in branch-     public static void main(String[] args)     {         System.out.println("Hello world");     }     public void methodInBranch()     {         System.out.println("Method in branch");     } }</pre>	<pre>public class Main {     //this is the chronological revision 1 in master     public static void main(String[] args)     {         System.out.println("Hello world");     } }</pre>	<pre>public class Main {     //this is the chronological revision 3 in master     public static void main(String[] args)     {         System.out.println("Hello world");     }     public void methodInMaster()     {         System.out.println("Method in master");     } }</pre>

# 3-way merge (Perforce P4Merge)

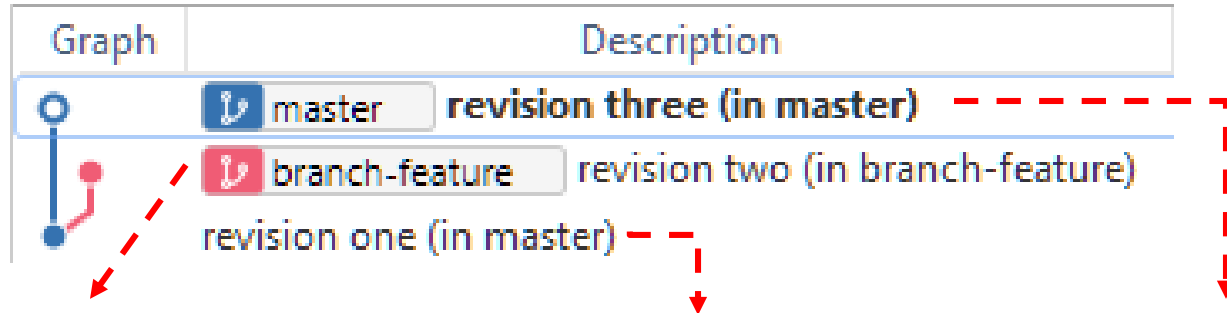


./Main_REMOTE_7736.java	./Main_BASE_7736.java	./Main_LOCAL_7736.java
<pre>public class Main {     //this is the chronological revision 2 in branch-     public static void main(String[] args)     {         System.out.println("Hello world");     }     public void methodInBranch()     {         System.out.println("Method in branch");     } }</pre>	<pre>public class Main {     //this is the chronological revision 1 in master     public static void main(String[] args)     {         System.out.println("Hello world");     } }</pre>	<pre>public class Main {     //this is the chronological revision 3 in master     public static void main(String[] args)     {         System.out.println("Hello world");     }     public void methodInMaster()     {         System.out.println("Method in master");     } }</pre>

Common ancestor (i.e., original version of the file before the changes in master and branch-feature)

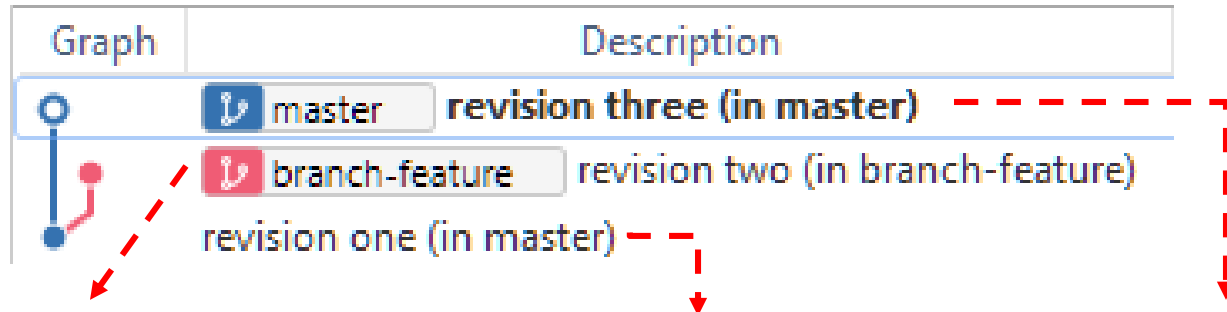
- Provides context of changes
- Makes 3-way merge less error-prone than 2-way merge

# 3-way merge (Perforce P4Merge)



./Main_REMOTE_7736.java	./Main_BASE_7736.java	./Main_LOCAL_7736.java
<pre>public class Main {     //this is the chronological revision 2 in branch-     public static void main(String[] args)     {         System.out.println("Hello world");     }     public void methodInBranch()     {         System.out.println("Method in branch");     } }</pre>	<pre>public class Main {     //this is the chronological revision 1 in master     public static void main(String[] args)     {         System.out.println("Hello world");     } }</pre>	<pre>public class Main {     //this is the chronological revision 3 in master     public static void main(String[] args)     {         System.out.println("Hello world");     }     public void methodInMaster()     {         System.out.println("Method in master");     } }</pre>

# 3-way merge (Perforce P4Merge)

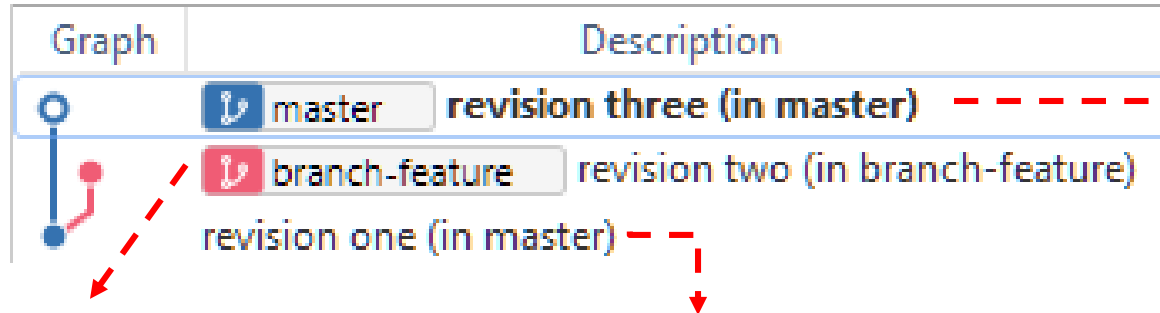


./Main_REMOTE_7736.java	./Main_BASE_7736.java	./Main_LOCAL_7736.java
<pre>public class Main {     //this is the chronological revision 2 in branch-     public static void main(String[] args)     {         System.out.println("Hello world");     }     public void methodInBranch()     {         System.out.println("Method in branch");     } }</pre>	<pre>public class Main {     //this is the chronological revision 1 in master     public static void main(String[] args)     {         System.out.println("Hello world");     } }</pre>	<pre>public class Main {     //this is the chronological revision 3 in master     public static void main(String[] args)     {         System.out.println("Hello world");     }     public void methodInMaster()     {         System.out.println("Method in master");     } }</pre>

Main.java
<pre>public class Main {     //this is the chronological revision 1 in master     //this is the chronological revision 2 in branch-feature     //this is the chronological revision 3 in master     public static void main(String[] args)     {         System.out.println("Hello world");     }     public void methodInBranch()     {         System.out.println("Method in branch");     }     public void methodInMaster()     {         System.out.println("Method in master");     } }</pre>

# 3-way merge (Perforce P4Merge)



./Main_REMOTE_7736.java	./Main_BASE_7736.java	./Main_LOCAL_7736.java
<pre>public class Main {     //this is the chronological revision 2 in branch-     public static void main(String[] args)     {         System.out.println("Hello world");     }     public void methodInBranch()     {         System.out.println("Method in branch");     } }</pre>	<pre>public class Main {     //this is the chronological revision 1 in master     public static void main(String[] args)     {         System.out.println("Hello world");     } }</pre>	<pre>public class Main {     //this is the chronological revision 3 in master     public static void main(String[] args)     {         System.out.println("Hello world");     }     public void methodInMaster()     {         System.out.println("Method in master");     } }</pre>

Main.java
<pre>public class Main {     //this is the chronological revision 1 in master     //this is the chronological revision 2 in branch-feature     //this is the chronological revision 3 in master     public static void main(String[] args)     {         System.out.println("Hello world");     }     public void methodInBranch()     {         System.out.println("Method in branch");     }     public void methodInMaster()     {         System.out.println("Method in master");     } }</pre>

Edit the final version of the merged changes (from master and/or branch-feature)



# 3-way merge (Perforce P4Merge)

Main.java - Perforce P4Merge

File Edit View Search Help

2 diffs (Ignore line ending differences) | Tab spacing: 4 | File format (Encoding: System Line endings: Windows)

Base: Main\_BASE\_7736.java  
Left: Main\_REMOTE\_7736.java  
Right: Main\_LOCAL\_7736.java

Differences from base: 0  
Differences from base: 0  
Conflicts: 2

Merge: Main.java

./Main_REMOTE_7736.java	./Main_BASE_7736.java	./Main_LOCAL_7736.java
<pre>public class Main {     //this is the chronological revision 2 in branch-     public static void main(String[] args)     {         System.out.println("Hello world");     }     public void methodInBranch()     {         System.out.println("Method in branch");     } }</pre>	<pre>public class Main {     //this is the chronological revision 1 in master     public static void main(String[] args)     {         System.out.println("Hello world");     } }</pre>	<pre>public class Main {     //this is the chronological revision 3 in master     public static void main(String[] args)     {         System.out.println("Hello world");     }     public void methodInMaster()     {         System.out.println("Method in master");     } }</pre>

Main.java

```
public class Main
{
    //this is the chronological revision 1 in master
    //this is the chronological revision 2 in branch-feature
    //this is the chronological revision 3 in master
    public static void main(String[] args)
    {
        System.out.println("Hello world");
    }
    public void methodInBranch()
    {
        System.out.println("Method in branch");
    }
    public void methodInMaster()
    {
        System.out.println("Method in master");
    }
}
```

# The default 3-way merge using vimdiff follows the same rationale/structure as P4Merge

```
public class Main
{
    //this is the chronological revision 3 in master

    public static void main(String[] args)
    {
        System.out.println("Hello world");
    }

    public void methodInMaster()
    {
        System.out.println("Method in master");
    }
}

public class Main
{
    //this is the chronological revision 1 in master

    public static void main(String[] args)
    {
        System.out.println("Hello world");
    }
}

public class Main
{
    //this is the chronological revision 2 in branch-feature

    public static void main(String[] args)
    {
        System.out.println("Hello world");
    }

    public void methodInBranch()
    {
        System.out.println("Method in branch");
    }
}

d/CS471/ExampleMergeConflict/Main_LOCAL_3028.java [dos] (13:39 28/02/2018) 3,1-2 All <S471/ExampleMergeConflict/Main_BASE_3028.java [dos] (13:39 28/02/2018) 3,1 All <geConflict/Main_REMOTE_3028.java [dos] (13:39 28/02/2018) 3,1 All
public class Main
{
    <<<<<< HEAD
    //this is the chronological revision 3 in master
    ||||| merged common ancestors
    //this is the chronological revision 1 in master
    =====
    //this is the chronological revision 2 in branch-feature
    >>>>>> branch-feature




    public static void main(String[] args)
    {
        System.out.println("Hello world");
    }

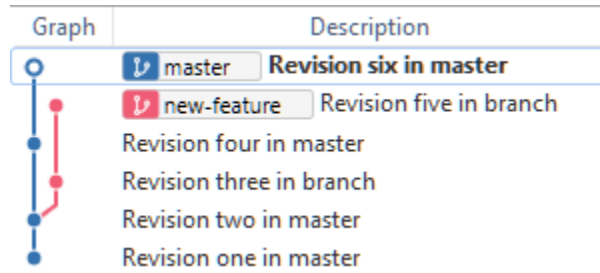
    public void methodInMaster()
    {
        System.out.println("Method in master");
    }

    ||||| merged common ancestors
    public void methodInBranch()
    {
        System.out.println("Method in branch");
    }
    >>>>>> branch-feature
}

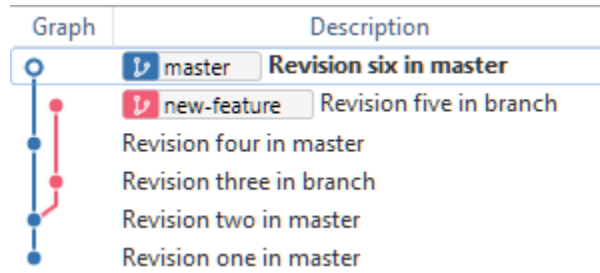
d/CS471/ExampleMergeConflict/Main.java [dos] (13:39 28/02/2018) 3,1-60 All
```

# Git Merge vs. Rebase Example

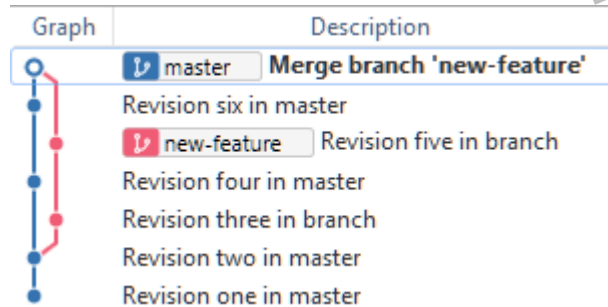
Graph	Description	
	 master	<b>Revision six in master</b>
	 new-feature	Revision five in branch
		Revision four in master
		Revision three in branch
		Revision two in master
		Revision one in master

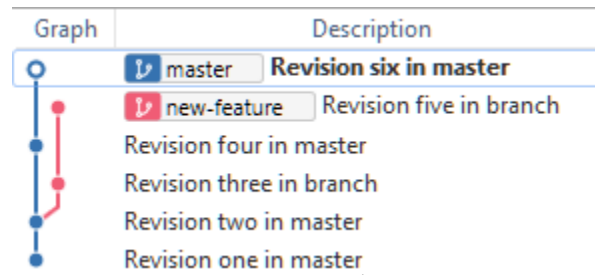


In master, run:  
`git merge new-feature`



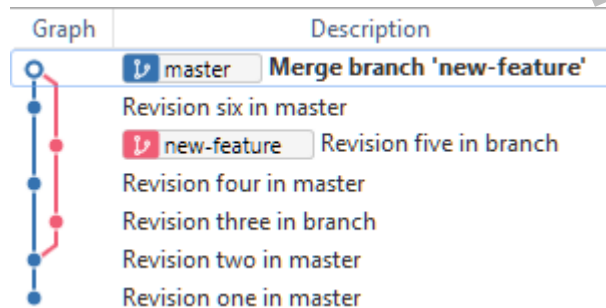
In master, run:  
`git merge new-feature`

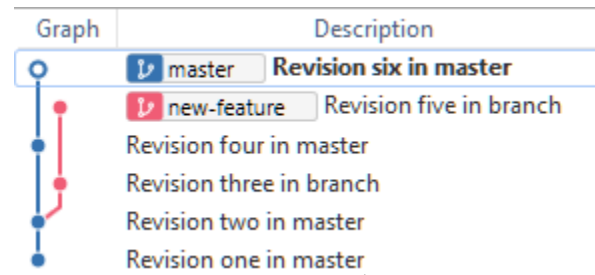




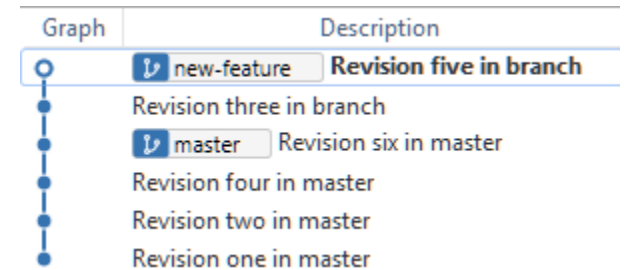
In branch, run:  
`git rebase master`

In master, run:  
`git merge new-feature`

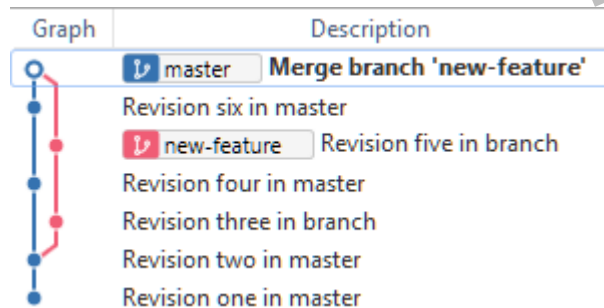




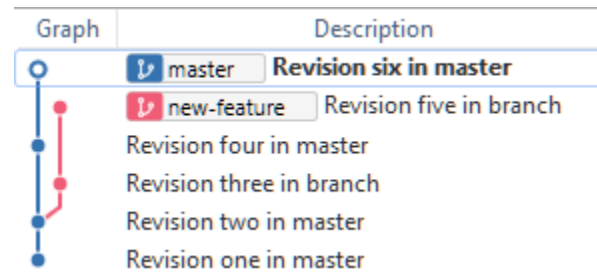
In branch, run:  
`git rebase master`



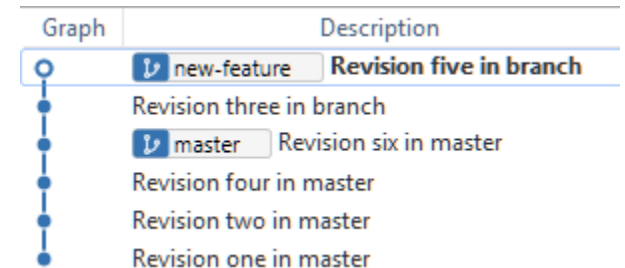
In master, run:  
`git merge new-feature`



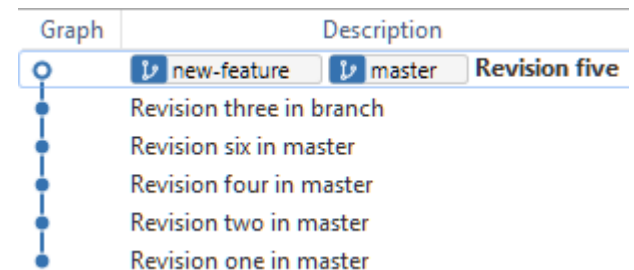




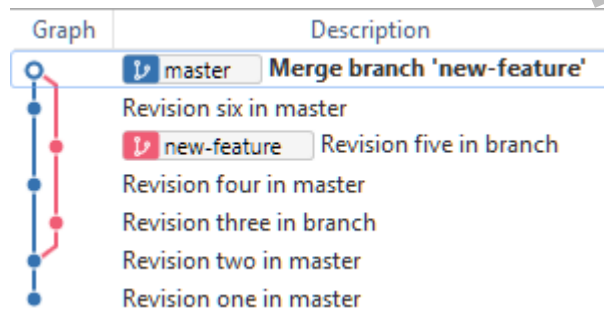
In branch, run:  
`git rebase master`

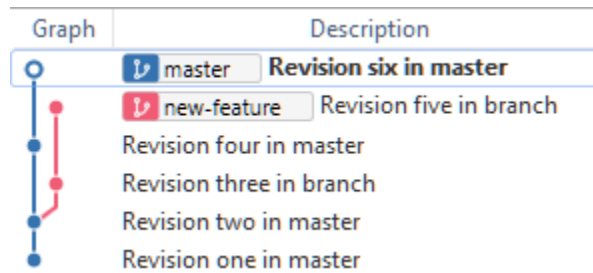


In master, run:  
`git merge new-feature`



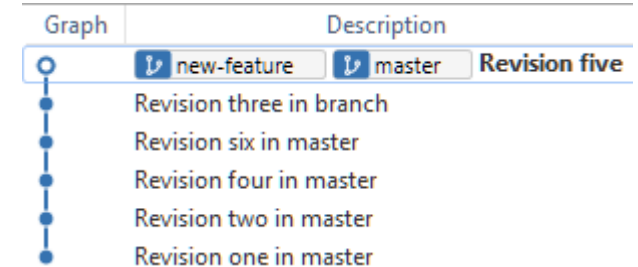
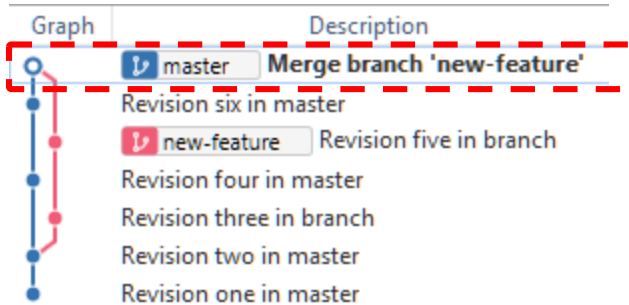
In master, run:  
`git merge new-feature`





merge

rebase/merge



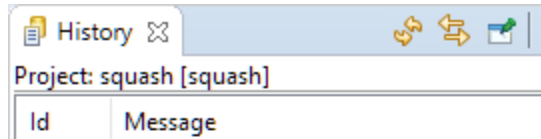
- Extra merge commit
- Non linear history
- + Preserves time information

- + No Extra commit
- + Linear history
- “Rewrites” history

# Git Squash Example

# Git Squash Example

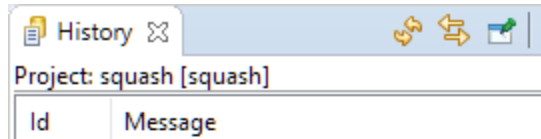
- Open the repository history in Eclipse (“Team” → “Show in History”)



79c259c fixes #345  
79bc243 implements #123

# Git Squash Example

- Open the repository history in Eclipse (“Team” → “Show in History”)

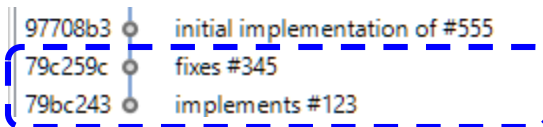
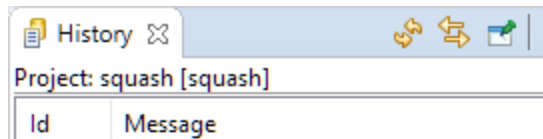


79c259c fixes #345  
79bc243 implements #123

Existing commits

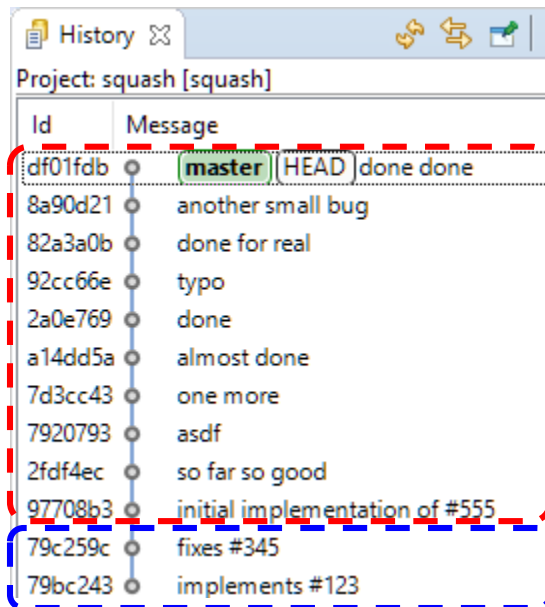
# Git Squash Example

- We start working on #555, and keep adding commits related to it



Existing commits

# Git Squash Example

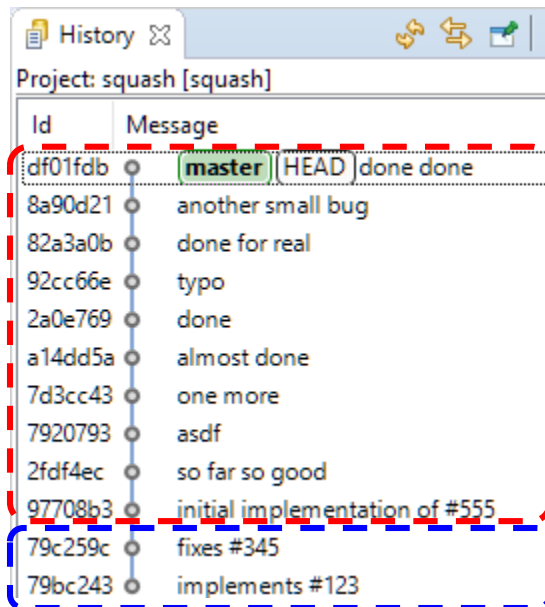


Id	Message
df01fdb	master HEAD done done
8a90d21	another small bug
82a3a0b	done for real
92cc66e	typo
2a0e769	done
a14dd5a	almost done
7d3cc43	one more
7920793	asdf
2fdf4ec	so far so good
97708b3	initial implementation of #555
79c259c	fixes #345
79bc243	implements #123

Multiple commits related to #555

Existing commits

# Git Squash Example



The screenshot shows a Git commit history window. The title bar says 'History' with a close button. Below it, the project name is 'Project: squash [squash]'. The table has two columns: 'Id' and 'Message'. The commits are listed in reverse chronological order. The top commit is 'df01fdb' with the message 'done done', which is the current HEAD and master branch. Below it are several commits related to issue #555, including 'another small bug', 'done for real', 'typo', 'done', 'almost done', 'one more', 'asdf', 'so far so good', and 'initial implementation of #555'. These are followed by 'fixes #345' and 'implements #123'. A red dashed box highlights the top commit and the series of commits for #555. A blue dashed box highlights the bottom two commits.

Id	Message
df01fdb	done done
8a90d21	another small bug
82a3a0b	done for real
92cc66e	typo
2a0e769	done
a14dd5a	almost done
7d3cc43	one more
7920793	asdf
2fdf4ec	so far so good
97708b3	initial implementation of #555
79c259c	fixes #345
79bc243	implements #123

## Multiple commits related to #555

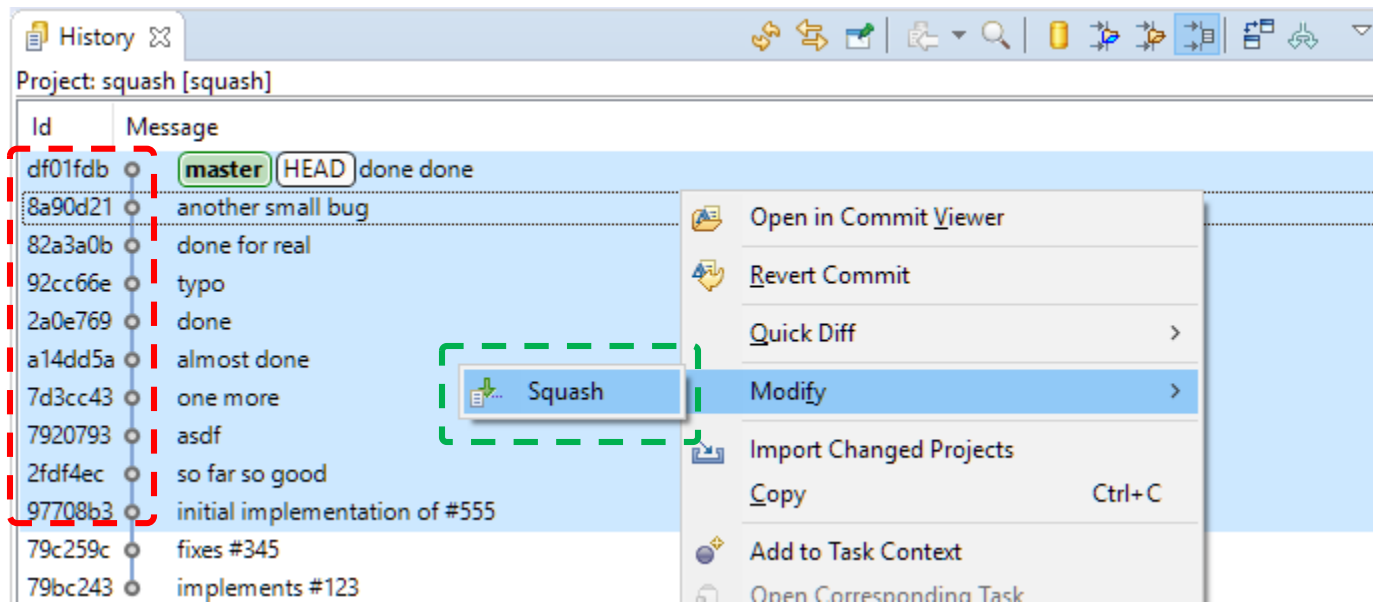
- Good for debugging/implementing
  - Allow “checkpoints”
- Bad for other team members to read
  - History clutter
- Solution: squash multiple related commits into one

Existing commits



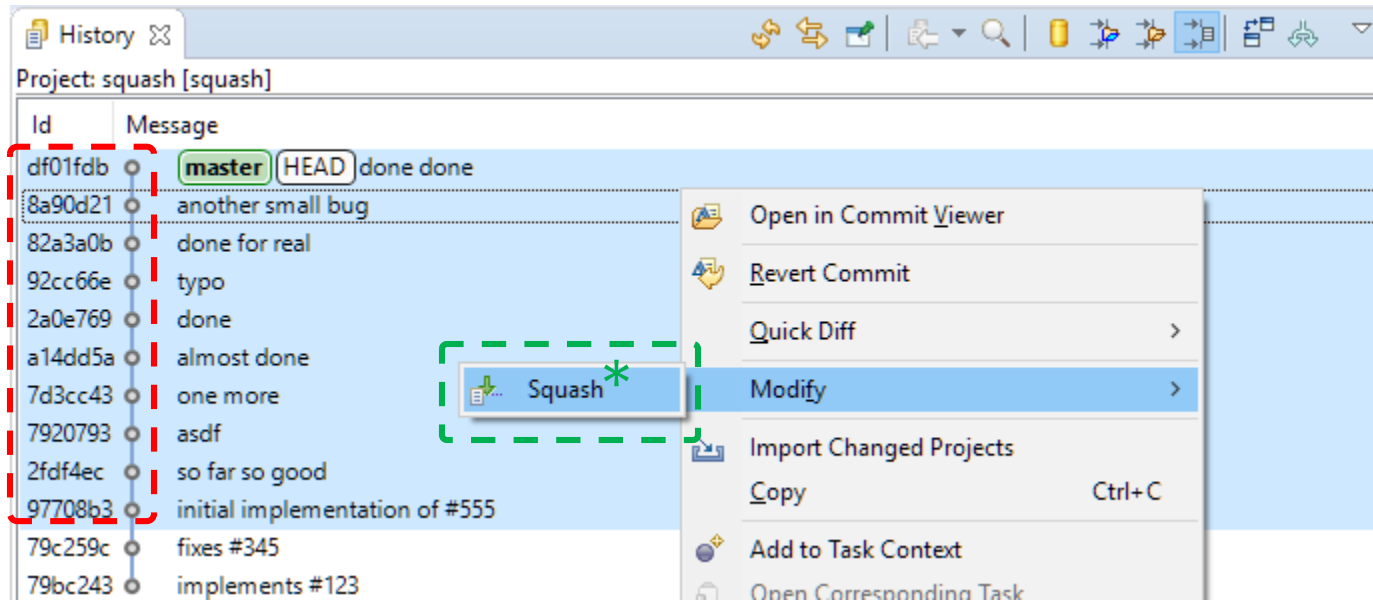
# Git Squash Example

- Select group of related commits that will be “compressed” / “squashed” into a new commit



# Git Squash Example

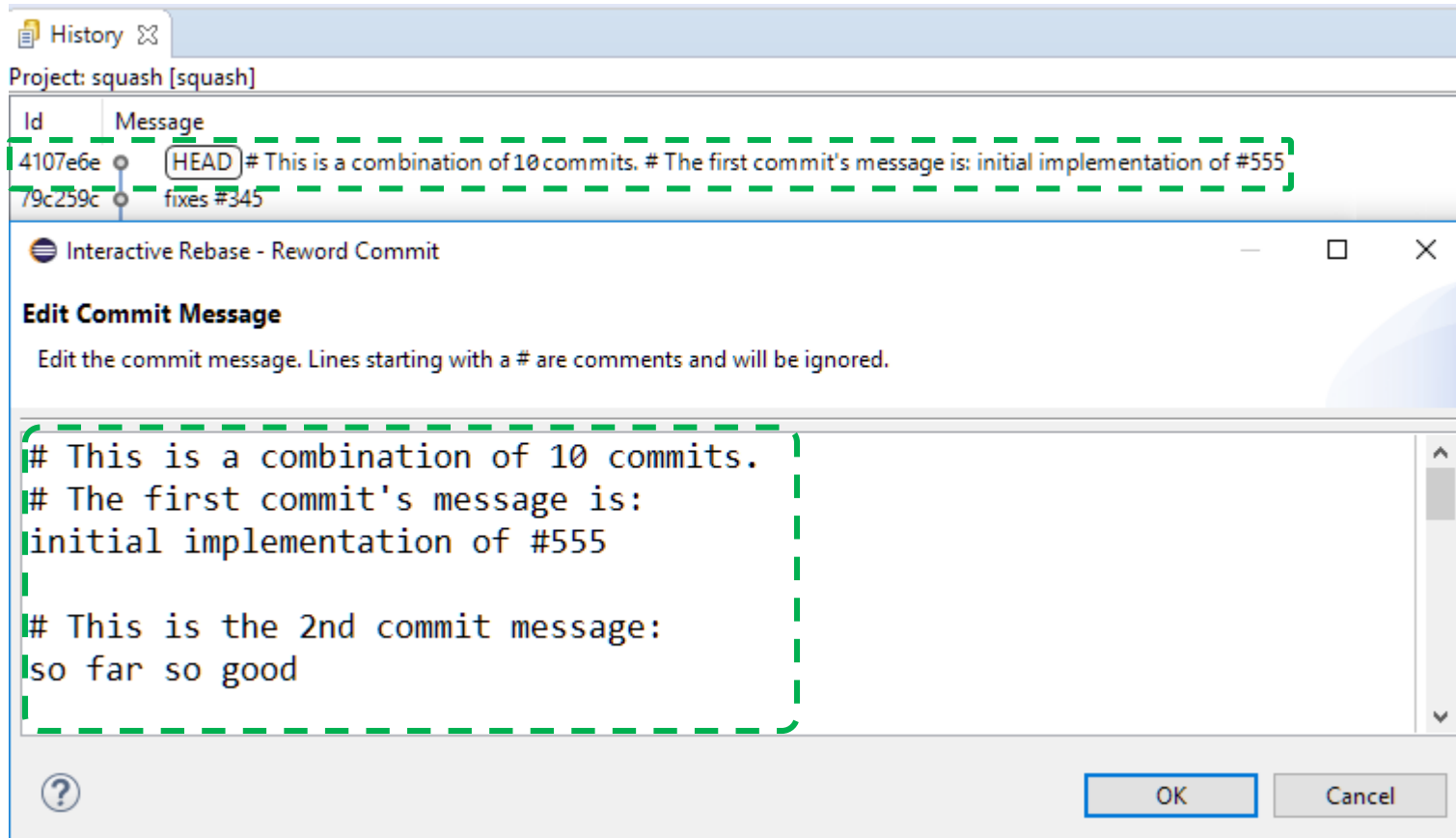
- Select group of related commits that will be “compressed” / “squashed” into a new commit



\* `git rebase -i HEAD~10`

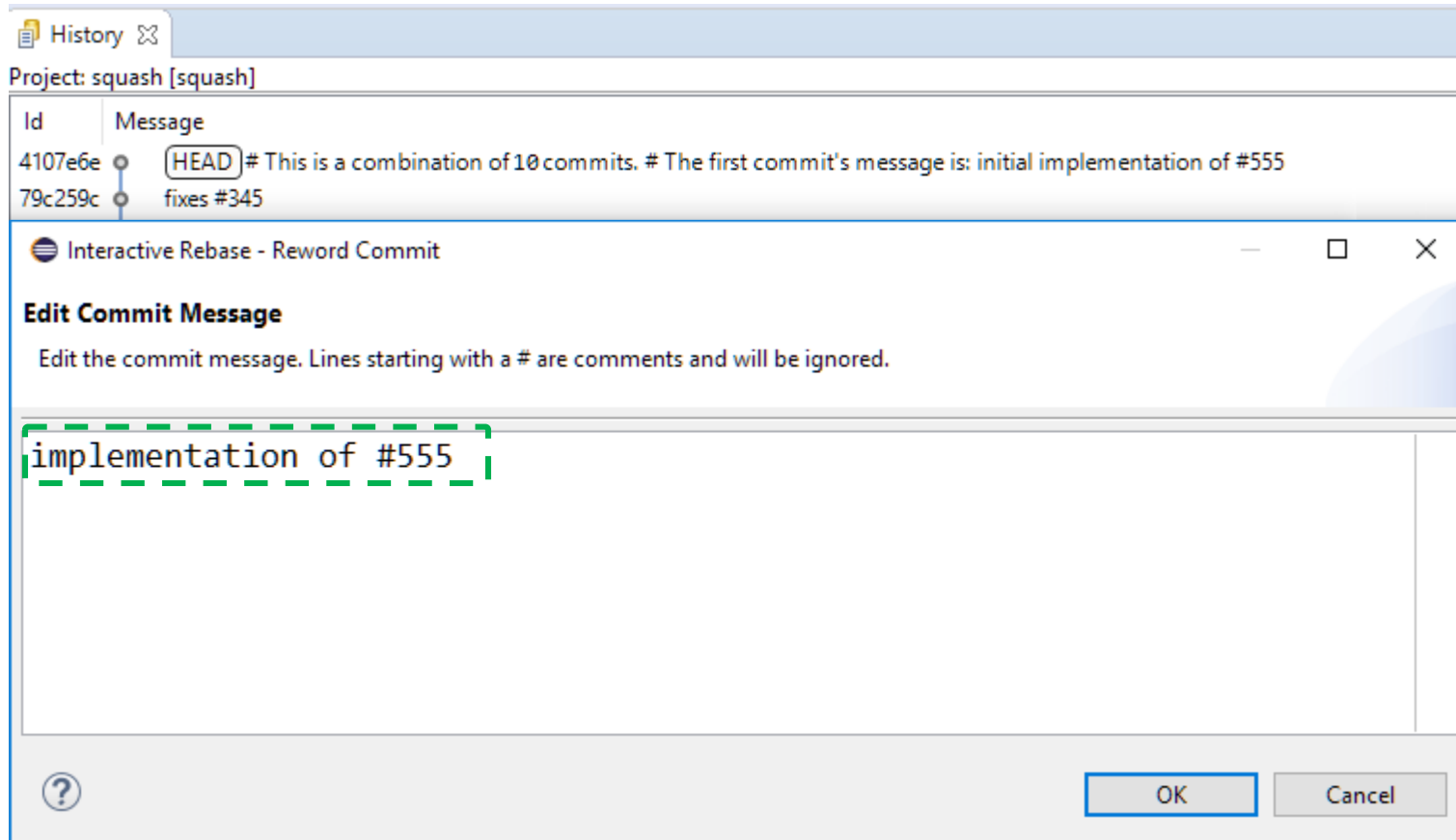
# Git Squash Example

- The 10 commits were squashed into one, and we can edit the commit log message of the resulting commit

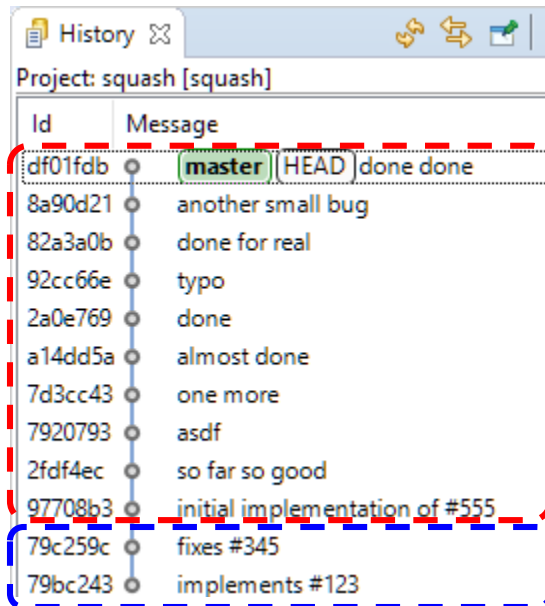


# Git Squash Example

- Updated commit log message



# Git Squash Example



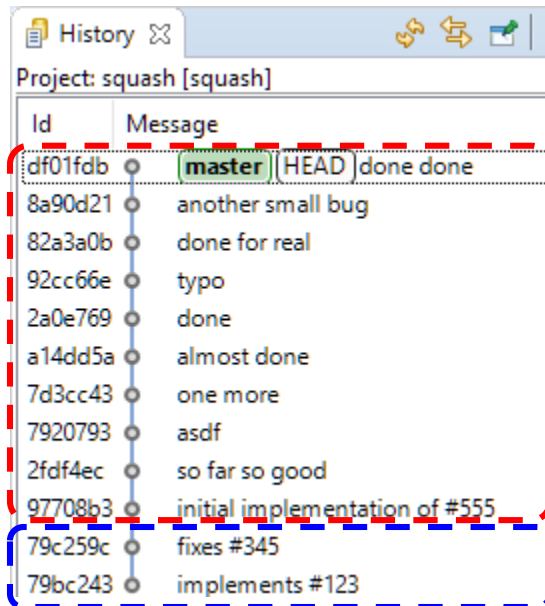
Id	Message
df01fdb	done done
8a90d21	another small bug
82a3a0b	done for real
92cc66e	typo
2a0e769	done
a14dd5a	almost done
7d3cc43	one more
7920793	asdf
2fdf4ec	so far so good
97708b3	initial implementation of #555
79c259c	fixes #345
79bc243	implements #123

Multiple commits  
related to #555

Existing commits

# Git Squash Example

- “squash”-ing works best on task branches and with pull-requests ↻

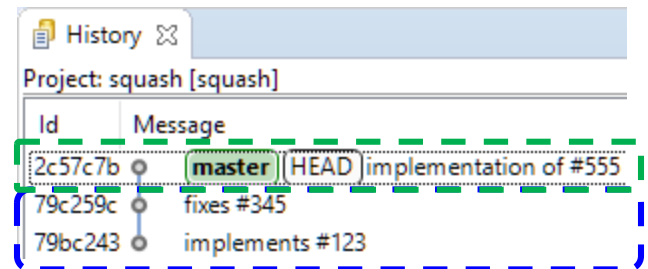


A screenshot of a Git commit history window for a project named 'squash'. The window shows a list of commits. The top commit, 'df01fdb', is the current HEAD and is labeled 'done done'. Below it are several other commits, including '8a90d21' (another small bug), '82a3a0b' (done for real), '92cc66e' (typo), '2a0e769' (done), 'a14dd5a' (almost done), '7d3cc43' (one more), '7920793' (asdf), '2fdf4ec' (so far so good), and '97708b3' (initial implementation of #555). The bottom two commits, '79c259c' (fixes #345) and '79bc243' (implements #123), are grouped together with a blue dashed box. A red dashed box encloses the top nine commits. A large grey arrow points from this window to the right window.

Id	Message
df01fdb	done done
8a90d21	another small bug
82a3a0b	done for real
92cc66e	typo
2a0e769	done
a14dd5a	almost done
7d3cc43	one more
7920793	asdf
2fdf4ec	so far so good
97708b3	initial implementation of #555
79c259c	fixes #345
79bc243	implements #123

Existing commits

Multiple commits  
related to #555  
were squashed into  
one commit



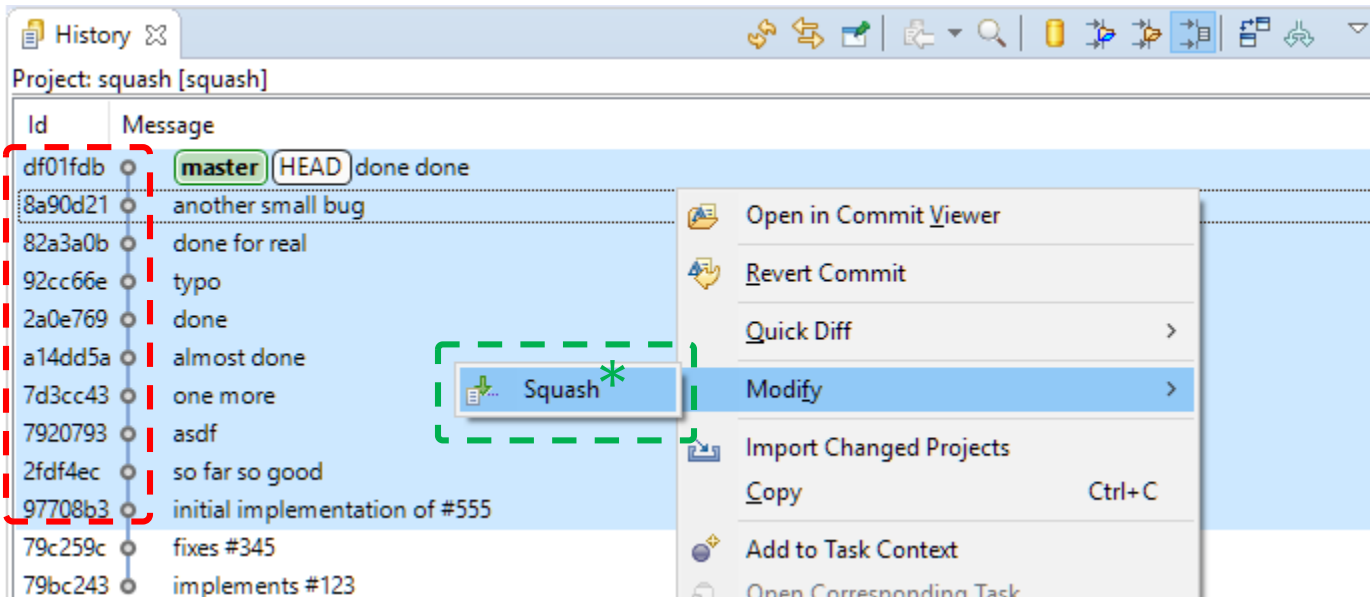
A screenshot of a Git commit history window for the same project 'squash' after a squash operation. The top commit, '2c57c7b', is the current HEAD and is labeled 'implementation of #555'. Below it are two other commits: '79c259c' (fixes #345) and '79bc243' (implements #123). The top commit is enclosed in a green dashed box, and the bottom two are enclosed in a blue dashed box.

Id	Message
2c57c7b	implementation of #555
79c259c	fixes #345
79bc243	implements #123

Existing commits

# Git Squash Example

## Eclipse GUI



\* `git rebase -i HEAD~10`

# Git Squash Example

Eclipse GUI

vs.

command line

The screenshot shows the Eclipse IDE's 'History' view on the left and a terminal window on the right. The 'History' view displays a list of commits for the 'squash' project. The commit 'df01fdb' is selected, and a 'Squash' button is highlighted with a green dashed box. The terminal window shows the output of a 'git rebase' command, which is a rebase of 10 commits onto the current HEAD. The terminal output includes a list of commands and their messages, followed by a summary of the rebase operation.

Id	Message
df01fdb	done done
8a90d21	another small bug
82a3a0b	done for real
92cc66e	typo
2a0e769	done
a14dd5a	almost done
7d3cc43	one more
7920793	asdf
2fdf4ec	so far so good
97708b3	initial implementation of #555
79c259c	fixes #345
79bc243	implements #123

```
pick 97708b3 initial implementation of #555
s 2fdf4ec so far so good
s 7920793 asdf
s 7d3cc43 one more
s a14dd5a almost done
s 2a0e769 done
s 92cc66e typo
s 82a3a0b done for real
s 8a90d21 another small bug
s df01fdb done done

# Rebase 79c259c..df01fdb onto 79c259c (10 commands)
#
# Commands:
# p, pick = use commit
# r, reword = use commit, but edit the commit message
# e, edit = use commit, but stop for amending
# s, squash = use commit, but meld into previous commit
# f, fixup = like "squash", but discard this commit's log message
# x, exec = run command (the rest of the line) using shell
# d, drop = remove commit
#
# These lines can be re-ordered; they are executed from top to bottom.
#
# If you remove a line here THAT COMMIT WILL BE LOST.
#
# However, if you remove everything, the rebase will be aborted.
#
# Note that empty commits are commented out
<cratch/squash/.git/rebase-merge/git-rebase-todo[+] [unix] (15:02 15/11/2017)10,2 A11
-- INSERT --
```

\* `git rebase -i HEAD~10`



# Pull Request Example

# In testPRBranch (created from master) make some minor changes to the file src/Main.java

The screenshot shows the GitHub web interface for a pull request. The top bar indicates the current branch is `testPRBranch`. The main area displays a diff for the file `src/Main.java`, showing changes between the `master` branch and the `testPRBranch`. The diff highlights a new method `demoMethodTask19()` added in the pull request, and a change to `demoMethodIllustrateCommitLink()`. The bottom section contains a summary and description area, and a button to `Commit to testPRBranch`.

Filter repositories

GitHub

CS481-Test

Other

Tutorial

testPRBranch

Changes History

Pull request

Update from master View branch Publish

1 change

src/Main.java

...	...	@@ -1,5 +1,10 @@
1	1	public class Main
2	2	{
	3 +	private void demoMethodTask19()
	4 +	{
	5 +	//implementation of task #19 in testPRBranch
	6 +	}
	7 +	
3	8	private void demoMethodIllustrateCommitLink()
4	9	{
5	10	//initial implementation of

Summary

Description

Commit to testPRBranch

# What would be a good commit log message?

- Ideally, we want to keep a “single source of truth”, namely the issue/task that the commit is supposed to implement
  - There is no need to add **duplicate** / **long** / **error-prone** / **insufficiently** detailed messages and commit descriptions (i.e., no need to “reinvent the wheel”)
- Solution: every commit should link to the ID of the issue / task / bug it supposed to implement / fix (e.g., “Implements #123”)

# In the commit log message specify the task/bug ID to associate the implementation with the task/bug it implements/fixes

The screenshot shows a GitHub Pull Request interface for a repository named "CS481-Test". The pull request is titled "testPRBranch" and is currently in the "Changes" tab. The "src/Main.java" file is selected, showing a diff with 1 change. The diff highlights the addition of a new method, `demoMethodTask19()`, which is associated with task #19. The commit message is being edited in the "Commit" section, with the text "implementation of task #" entered. A dropdown menu is visible, showing a list of tasks to select from: #19 Sample Task to Illustrate a Pull Request, #17 Task 4 Story 1, #1 Create account, #15 Customize ZenHub, and #11 Story Six. The "Commit" button is highlighted with a red dashed box.

Filter repositories

GitHub

CS481-Test

Other

Tutorial

testPRBranch

Update from master

View branch

1 change

src/Main.java

```
... 1 @@ -1,5 +1,10 @@
... 2 public class Main
... 3 {
... 4     private void demoMethodTask19()
... 5     {
... 6         //implementation of task #19 in testPRBranch
... 7     }
... 8     private void demoMethodIllustrateCommitLink()
... 9     {
... 10         //initial implementation of
```

implementation of task #

Description

- #19 Sample Task to Illustrate a Pull Request
- #17 Task 4 Story 1
- #1 Create account
- #15 Customize ZenHub
- #11 Story Six

Commit

# Prepare Pull Request

The screenshot shows the GitHub web interface for a repository named 'CS481-Test'. The current branch is 'testPRBranch'. A pull request is being prepared, titled 'implementation of task #19' by user 'bgdndit'. The pull request is linked to the 'master' branch. The code diff shows changes to 'src/Main.java', specifically adding a 'demoMethodTask19()' method. The 'Pull request' button is highlighted with a red dashed box. The 'Publish' button is also visible. The 'Changes' and 'History' tabs are at the top. The 'testPRBranch' is highlighted with a red dashed box. The code diff shows changes to 'src/Main.java', specifically adding a 'demoMethodTask19()' method. The 'Pull request' button is highlighted with a red dashed box. The 'Publish' button is also visible. The 'Changes' and 'History' tabs are at the top. The 'testPRBranch' is highlighted with a red dashed box.

Filter repositories

GitHub

CS481-Test

Other

Tutorial

testPRBranch

Update from master

View branch

Changes

History

Pull request

Publish

implementation of task #19  
just now by bgdndit

implementation of task #19

bgdndit 04d51ef

GitHub Revert Collapse all

src/Main.java

```
@@ -1,5 +1,10 @@
1 1 public class Main
2 2 {
3 + private void demoMethodTask19()
4 + {
5 +     //implementation of task #19 in testPRBranch
6 + }
7 +
3 8 private void demoMethodIllustrateCommitLink()
4 9 {
5 10     //initial implementation of
```

Change made to  
local testPRBranch

# Description of PR should include a reference to the task ID\*

implementation of task #19

1 minute ago by bgndnit

implementation of task #19

src/Main.java

```
... 1 @@ -1,5 +1,10 @@
... 2 public class Main
... 3 {
... 4     private void
... 5     demoMethodTask19()
... 6     {
... 7         //implementation of
... 8         task #19 in
... 9         testPRBranch
... 10    }
... 11    private void
... 12    demoMethodIllustrateCommit
... 13    Link()
... 14    {
... 15        //initial
... 16        implementation of
```

This pull request can be automatically merged.

☒ Send pull request

Your local commits will be synced to GitHub.

\*Reference to the task ID in the title of the PR is not sufficient to create the link (traceability) between the task and the PR in GitHub

# A new Pull Request (#20) was created

Filter repositories

GitHub

CS481-Test

Other

Tutorial

testPRBranch

ChangesHistory

#20

Pull request

implementation of task #19  
2 minutes ago by bgnddit

implementation of task #19  
bgnddit 04d51ef GitHub Revert Cc

src\Main.java

...	...	@@ -1,5 +1,10 @@
1	1	public class Main
2	2	{
	3 +	private void demoMethodTask19()
	4 +	{
	5 +	//implementation of task #19 in testPRBranch
	6 +	}
	7 +	
3	8	private void demoMethodIllustrateCommitLink()
4	9	{
5	10	//initial implementation of



Pull request created 🎵💖

[View it on GitHub](#)

# Why is Traceability Important?

- Establishing a traceability (link) between **task #19** and **PR #20**
  - allows developers who view **task #19** to navigate / investigate **PR #20** (containing the commits and the changed code)
  - (vice-versa) allows developers to navigate from **PR #20** back to **task #19**, in order to understand the context / requirement for the change
    - **task #19** should further link to the a user story, for additional context
- Note: Traceability will be heavily emphasized in the group project



# Task #19 is linked to the PR #20

The screenshot shows a GitHub issue page for "Sample Task to Illustrate a Pull Request #19". The issue is open and was created 9 minutes ago by user bgdndit. The issue description states it is a sample task linked to story #1, illustrating a pull request linked to a task, which is already linked to a story to show full traceability of software artifacts. The "TODO:" section lists three tasks: creating a new branch called testPRBranch, making changes to src/Main.java, and issuing a pull request to merge testPRBranch to the master branch. The right sidebar shows the issue's metadata, including Pipeline (New Issues), Assignees (bgdndit), Labels (None yet), Projects (None yet), Milestone (Sprint 1), Estimate (1), Epics (Not inside an Epic), and Notifications (Unsubscribe). The bottom of the page shows a commit history for the issue, with a commit titled "implementation of task #19" by bgdndit, which is highlighted with a red dashed box. The commit message "implementation of task #19" is also highlighted with a red dashed box, and a green "Open" button is visible next to it. The browser address bar shows the URL "https://github.com/BoiseState/CS481-Test/issues/19".

Sample Task to Illustrate a Pull Request #19

Open bgdndit opened this issue 9 minutes ago · 0 comments

bgdndit commented 9 minutes ago

This is a sample task linked to story #1 that will illustrate a pull request linked to a task, which is already linked to a story to illustrate the full traceability of these software artifacts.

TODO:

- ☐ Create a new branch in the local git repository called `testPRBranch`
- ☐ In `testPRBranch` make some minor changes to the file `src/Main.java`
- ☐ From the GitHub desktop application issue a pull request to merge `testPRBranch` to the `master` branch in the remote repository

bgdndit added this to the **Sprint 1** milestone 9 minutes ago

bgdndit self-assigned this 9 minutes ago

bgdndit set the estimate to **1** 9 minutes ago

bgdndit referenced this issue a minute ago

implementation of task #19 #20

Open

bgdndit added a commit that referenced this issue a minute ago

implementation of task #19 04d51ef

**Pipeline**

New Issues

**Assignees**

bgdndit

**Labels**

None yet

**Projects**

None yet

**Milestone**

Sprint 1

**Estimate**

1

**Epics**

Not inside an Epic

**Notifications**

Unsubscribe

You're receiving notifications because you were assigned.

# Task #19 is linked to the PR #20

The screenshot shows a GitHub issue page for 'Sample Task to Illustrate a Pull Request #19'. The issue is open and was created 9 minutes ago by user 'bgdndit'. The issue description states: 'This is a sample task linked to story #1 that will illustrate a pull request linked to a task, which is already linked to a story to illustrate the full traceability of these software artifacts.' Below the description is a 'TODO:' section with three tasks: 'Create a new branch in the local git repository called testPRBranch', 'In testPRBranch make some minor changes to the file src/Main.java', and 'From the GitHub desktop application issue a pull request to merge testPRBranch to the master branch in the remote repository'. The issue is linked to a milestone 'Sprint 1' and has an estimate of 1. The issue is also linked to a pull request 'implementation of task #19 #20', which is highlighted with a red dashed box. The pull request is open and has a green 'Open' button. The right sidebar shows the 'Pipeline' section with 'New Issues' and 'Assignees' (bgdndit). The 'Labels' section is empty. The 'Projects' section is empty. The 'Milestone' section shows 'Sprint 1' with a progress bar. The 'Estimate' section shows '1'. The 'Epics' section is empty.

Sample Task to Illustrate a Pull Request #19

Open bgdndit opened this issue 9 minutes ago · 0 comments

bgdndit commented 9 minutes ago

This is a sample task linked to story #1 that will illustrate a pull request linked to a task, which is already linked to a story to illustrate the full traceability of these software artifacts.

TODO:

- ☐ Create a new branch in the local git repository called `testPRBranch`
- ☐ In `testPRBranch` make some minor changes to the file `src/Main.java`
- ☐ From the GitHub desktop application issue a pull request to merge `testPRBranch` to the `master` branch in the remote repository

bgdndit added this to the **Sprint 1** milestone 9 minutes ago

bgdndit self-assigned this 9 minutes ago

bgdndit set the estimate to **1** 9 minutes ago

bgdndit referenced this issue a minute ago

implementation of task #19 #20

Open

**Pipeline**

New Issues

**Assignees**

bgdndit

**Labels**

None yet

**Projects**

None yet

**Milestone**

Sprint 1

**Estimate**

1

**Epics**

Not inside an Epic

Allows other developers to observe the status of task #19 and to navigate to the source code of PR #20 if needed

The screenshot shows a web browser window displaying the GitHub interface for the repository 'BoiseState / CS481-Test'. The page title is 'Pull Requests · BoiseState'. The browser address bar shows the URL 'https://github.com/BoiseState/CS481-Test/pulls'. The repository is marked as 'Private'. The page has navigation tabs for 'Code', 'Issues 12', 'Pull requests 1', 'Boards', 'Reports', 'Projects 0', and 'Wiki'. A search bar with the filter 'is:pr is:open' is present. A green button labeled 'New pull request' is in the top right. Below the navigation, there is a summary bar showing '1 Open' and '2 Closed' pull requests. A red dashed box highlights a pull request titled 'implementation of task #19', which was opened 2 minutes ago by user 'bgndnit'. Below this, a 'ProTip!' suggests typing 'g i' on any issue or pull request to go back to the issue listing page. The footer contains copyright information for 2017 GitHub, Inc., and links to 'Terms', 'Privacy', 'Security', 'Status', 'Help', 'Contact GitHub', 'API', 'Training', 'Shop', 'Blog', and 'About'.

Pull Requests · BoiseState

GitHub, Inc. [US] | https://github.com/BoiseState/CS481-Test/pulls

This repository Search Pull requests Issues Gist ToDo

BoiseState / CS481-Test Private

Unwatch 9 Unstar 1 Fork 0

Code Issues 12 Pull requests 1 Boards Reports Projects 0 Wiki

Filters is:pr is:open Labels Milestones New pull request

1 Open 2 Closed Author Labels Milestones Assignee Sort

implementation of task #19  
#20 opened 2 minutes ago by bgndnit New Issues

ProTip! Type g i on any issue or pull request to go back to the issue listing page.

© 2017 GitHub, Inc. Terms Privacy Security Status Help Contact GitHub API Training Shop Blog About

Alternative way to access PRs (i.e., see all open PRs that other developers can contribute to...)

The screenshot shows a GitHub pull request page. The repository is 'BoiseState / CS481-Test'. The pull request is titled 'implementation of task #19' and is from the 'testPRBranch' to the 'master' branch. The pull request is created by 'bgndnit' and is currently open. The commit message is 'implementation of task #19'. The pull request has 1 commit and 1 file changed. The status is 'This branch has no conflicts with the base branch'. The 'Merge pull request' button is visible. A red dashed box highlights the commit message 'implementation of task #19' in the comment section.

## implementation of task #19

- implementation of task #19

Add more commits by pushing to the **testPRBranch** branch on **BoiseState/CS481-Test**.

 This branch has no conflicts with the base branch  
Merging can be performed automatically.

## Merge pull request

You can also [open this in GitHub Desktop](#) or view [command line instructions](#).

## Pipeline

## New Issues

## Reviewers

No reviews—request one

### Assignees

No one—assign yourself

### Labels

None yet

## Projects

None yet

### Milestone

No milestone

# PR #20 referencing task #19

The screenshot shows a GitHub pull request page for the repository 'BoiseState / CS481-Test'. The pull request is titled 'implementation of task #19' and is from the 'testPRBranch' to the 'master' branch. A red dashed box highlights the title 'implementation of task #19' and a comment by 'bgdndit' that says 'implementation of task #19'. The comment is 3 minutes old. The pull request has 1 commit and 1 file changed. The right sidebar shows the 'Pipeline' and 'Reviewers' sections. A yellow callout box with a red border is overlaid on the bottom right of the page.

implementation of task #19

Open bgdndit wants to merge 1 commit into master from testPRBranch

Conversation 0 Commits 1 Files changed 1 +5 -0

bgdndit commented 3 minutes ago

implementation of task #19

implementation of task #19 04d51ef

No reviews—request one

Allows developers to navigate from PR #20 back to task #19, in order to understand the context / requirement for the change

# PR #20 referencing task #19

The screenshot shows a GitHub pull request interface for the repository **BoiseState / CS481-Test**. The pull request is titled **implementation of task #19** and is currently open. The pull request is from the **testPRBranch** to the **master** branch. The pull request is created by **bgdndit** and wants to merge 1 commit into **master** from **testPRBranch**. The pull request is currently open and has 1 commit and 1 file changed.

A comment by **bgdndit** is visible, stating "commented 3 minutes ago" and "implementation of task #19".

The pull request details show the commit hash **04d51ef** and the file **implementation of task #19**. The code diff shows the following changes:

```
@@ -1,5 +1,10 @@
public class Main
{
    private void
    demoMethodTask19()
    {
        //implementation of
        task #19 in
        testPRBranch
    }
    private void
```

The pull request status indicates that the branch has no conflicts with the base branch and can be merged automatically. The **Send pull request** button is visible, and a message states: "Your local commits will be synced to GitHub."

Red dashed boxes highlight the pull request title "implementation of task #19" and the comment "implementation of task #19". A red dashed line connects the comment to the pull request title.



The screenshot shows a GitHub pull request page for the repository 'BoiseState / CS481-Test'. The pull request is titled 'implementation of task #19' and is from the 'testPRBranch' to the 'master' branch. A yellow callout box with a red border contains the text: 'Subtle detail for accessibility: hyperlink #19 is clickable'. This box points to the 'implementation of task #19' link in the pull request title and the 'implementation of task #19' link in the pull request description. The pull request description also contains a code block with the following content:

```
5 +1,10 @@
class Main

private void
demoMethodTask19()
{
    //implementation of
    task #19 in
    testPRBranch
}

private void
```

The pull request status shows 'This branch has no conflicts with the base branch' and 'Merging can be performed automatically.' The 'Send pull request' button is visible at the bottom right of the pull request card.

Subtle detail for accessibility:  
hyperlink #19 is clickable

# PR #20 referencing task #19

Whereas the #19 in the title is not clickable

implementation of task #19#20

bgdndit commented 3 minutes ago

implementation of task #19

Subtle detail for accessibility:  
hyperlink #19 is clickable

✓ This branch has no conflicts with the base branch  
Merging can be performed automatically.

Merge pull request

You can also open this in GitHub Desktop

ation of task #19

04d51ef GitHub Revert Cc

```
5 +1,10 @@
class Main
```

```
private void
demoMethodTask19()
{
    //implementation of
    task #19 in
    testPRBranch
}
```

```
private void
```

Pull request

from testPRBranch into master

implementation of task #19

implementation of task #19

✓ This pull request can be automatically merged.

✓ Send pull request

Your local commits will be synced to GitHub.

Milestone

No milestone



# PR allows you to Request Reviewer

The screenshot shows a GitHub web interface for a pull request titled "implementation of task #19 #20". The repository is "BoiseState / CS481-Test". The pull request is from branch "testPRBranch" to "master", containing 1 commit and 1 file change. A comment from "bgdndit" is visible. A status bar indicates "This branch has no conflicts with the base branch" and provides a "Merge pull request" button. A "Request a review" modal is open, showing a search for "shane" and a selection of "shane" (Shane K. Panter). The modal is highlighted with a red dashed border. The right sidebar shows options for Pipeline, Reviewers, Labels, Projects, and Milestone.

implementation of task #19 #20

Open bgdndit wants to merge 1 commit into master from testPRBranch

Conversation 0 Commits 1 Files changed 1 +5 -0

bgdndit commented 5 minutes ago

implementation of task #19

implementation of task #19

Add more commits by pushing to the testPRBranch branch on BoiseState/CS481-Test.

✓ This branch has no conflicts with the base branch  
Merging can be performed automatically.

Merge pull request You can also open this in GitHub Desktop or view command line instructions.

Write Preview

Request a review

shane

shane Shane K. Panter

Labels  
None yet

Projects  
None yet

Milestone  
No milestone

implementation of task #19 #20

Open bgnddit wants to merge 1 commit into master from testPRBranch

Conversation 0 Commits 1 Files changed 1 +5 -0

bgnddit commented 6 minutes ago

implementation of task #19

implementation of task #19 @4d51ef

bgnddit requested a review from shanep 18 seconds ago

bgnddit self-assigned this just now

bgnddit added this to the Sprint 1 milestone just now

Add more commits by pushing to the testPRBranch branch on BoiseState/CS481-Test.

This branch has no conflicts with the base branch  
Merging can be performed automatically.

Merge pull request You can also open this in GitHub Desktop or view command line instructions.

Assignees

bgnddit

Labels

None yet

Projects

None yet

Milestone

Sprint 1

Estimate

No estimate yet

Add more commits by pushing to the `testPRBranch` branch on BoiseState/CS481-Test.

# Leave feedback for the change

The screenshot shows a GitHub pull request page for the repository `BoiseState/CS481-Test`. The pull request is titled "implementation of task #19" and is currently in the "Open" state. The commit hash is `04d51ef`.

The commit history shows the following actions:

- `implementation of task #19` (commit `04d51ef`)
- `bgdndit` requested a review from `shanep` a minute ago
- `bgdndit` self-assigned this 38 seconds ago
- `bgdndit` added this to the **Sprint 1** milestone 25 seconds ago

A message indicates: "Add more commits by pushing to the **testPRBranch** branch on **BoiseState/CS481-Test**."

A green box with a checkmark states: "This branch has no conflicts with the base branch. Merging can be performed automatically." Below this is a green button labeled "Merge pull request" and a link to "view command line instructions".

The right sidebar shows the following sections:

- New Issues**
- Reviewers**: `shanep`
- Assignees**: `bgdndit`
- Labels**: None yet
- Projects**: None yet
- Milestone**: **Sprint 1**
- Estimate**: No estimate yet
- Epics**: Not inside an Epic
- Notifications**: Unsubscribe button. Text: "You're receiving notifications because you were assigned."
- 1 participant**

The comment box at the bottom is highlighted with a red dashed border. It contains the text "leave feedback for the change". Below the text box is a link: "Attach files by dragging & dropping, selecting them, or pasting from the clipboard." At the bottom of the comment box are two buttons: "Close and comment" and "Comment".

# Reviewer can inspect the code

The screenshot shows a GitHub pull request interface. At the top, the title is "implementation of task #19 #20". Below the title, it says "bgnddit wants to merge 1 commit into master from testPRBranch". The interface includes tabs for "Conversation" (0), "Commits" (1), and "Files changed" (1). A green "Open" button is visible. The main content area shows a commit titled "implementation of task #19" with a commit hash of "04d51ef". A red dashed box highlights this commit. Below the commit, there are several status updates: "bgnddit commented 8 minutes ago" with the comment "implementation of task #19", "bgnddit requested a review from shanep 2 minutes ago", "bgnddit self-assigned this 2 minutes ago", and "bgnddit added this to the Sprint 1 milestone a minute ago". A second comment box shows "bgnddit commented just now" with the comment "leave feedback for the change". On the right side, there are sections for "Pipeline", "Reviewers" (showing shanep), "Assignees" (showing bgnddit), "Labels", "Projects", "Milestone" (Sprint 1), "Estimate", and "Epics". At the bottom, there are two green checkmark icons with the text "Review requested" and "This branch has no conflicts with the base branch".

implementation of task #19 #20

Open bgnddit wants to merge 1 commit into master from testPRBranch

Conversation 0 Commits 1 Files changed 1 +5 -0

bgnddit commented 8 minutes ago

implementation of task #19

implementation of task #19 04d51ef

bgnddit requested a review from shanep 2 minutes ago

bgnddit self-assigned this 2 minutes ago

bgnddit added this to the Sprint 1 milestone a minute ago

bgnddit commented just now

leave feedback for the change

Add more commits by pushing to the testPRBranch branch on BoiseState/CS481-Test.

Review requested  
Review has been requested on this pull request. It is not required to merge. [Learn more.](#)

This branch has no conflicts with the base branch

Pipeline

New Issues

Reviewers

shanep

Assignees

bgnddit

Labels

None yet

Projects

None yet

Milestone

Sprint 1

Estimate

No estimate yet

Epics

Not inside an Epic

# Inspect code

implementation of task #

GitHub, Inc. [US] | https://github.com/BoiseState/CS481-Test/pull/20/commits/04d51ef8cd13e552a58144e78128ca33857243e3

implementation of task #19 #20


Open bgdndit wants to merge 1 commit into master from testPRBranch

Conversation 1 Commits 1 Files changed 1

Changes from all commits 1 file +5 -0

Unified Split Review changes

implementation of task #19

 bgdndit committed 12 minutes ago commit 04d51ef8cd13e552a58144e78128ca33857243e3

5 src/Main.java

... @@ -1,5 +1,10 @@

1 public class Main

2 {

3 private void demoMethodIllustrateCommitLink()

4 {

5 //initial implementation of

1 public class Main

2 {

3 + private void demoMethodTask19()

4 + {

5 + //implementation of task #19 in testPRBranch

6 + }


7 +

8 private void demoMethodIllustrateCommitLink()

9 {

10 //initial implementation of

© 2017 GitHub, Inc. Terms Privacy Security Status Help



Contact GitHub API Training Shop Blog About

# Leave feedback directly in the code...

implementation of task #19

bgdndit committed 12 minutes ago

commit 04d51ef8cd13e552a58144e78128ca33857243e3

5 src/Main.java

```
1 public class Main
2 {
3 + private void demoMethodTask19()
4 + {
5 + //implementation of task #19 in testPRBranch
6 + }
7 +
8 private void demoMethodIllustrateCommitLink()
```

Write Preview **AA** *B* *i* “ < > ↺ ⋮ ⋮ ⋮ ↶ @ 📌

leave feedback directly in the code...

Attach files by dragging & dropping, selecting them, or pasting from the clipboard.

Styling with Markdown is supported

Cancel Add single comment Start a review

# Starting a review...

The screenshot shows a GitHub pull request page for the repository 'BoiseState/CS481-Test'. The pull request is titled 'implementation of task #19' and was committed by user 'bgdndit' 13 minutes ago. The commit hash is 04d51ef8cd13e552a58144e78128ca33857243e3. The interface shows a diff view for the file 'src/Main.java'. The changes include adding a new method 'demoMethodTask19()' and updating 'demoMethodIllustrateCommitLink()'. A review comment from 'bgdndit' is visible, stating 'leave feedback directly in the code...'. The interface includes buttons for 'Unified', 'Split', 'Review changes 1', 'Show comments', 'View', 'Start a new conversation', and 'Finish your review'.

implementation of task #19

bgdndit committed 13 minutes ago

commit 04d51ef8cd13e552a58144e78128ca33857243e3

5 src/Main.java

@@ -1,5 +1,10 @@

```
1 public class Main
2 {
3     private void demoMethodTask19()
4     {
5         //implementation of task #19 in testPRBranch
6     }
7 }
8 private void demoMethodIllustrateCommitLink()
9 {
10     //initial implementation of
```

bgdndit just now Pending

leave feedback directly in the code...

Reply...

Start a new conversation Finish your review

# Continue with the review

The screenshot shows a GitHub pull request review interface. At the top, the browser address bar displays the URL: <https://github.com/BoiseState/CS481-Test/pull/20/commits/04d51ef8cd13e552a58144e78128ca33857243e3>. The page header indicates "Changes from all commits" with "1 file" and a diff summary of "+5 -0". A green button labeled "Review changes 1" is visible. The commit is attributed to "bganait" committed "14 minutes ago".

The code diff for `src/Main.java` shows the following changes:

```
1 public class Main
2 {
3 +   private void demoMethodTask19()
4 +   {
5 +       //implementation of task #19 in testPRBranch
```

A review comment by user "bgdndit" (29 seconds ago, Pending) is shown with the text "leave feedback directly in the code...". Below the comment is a rich text editor with a toolbar. The editor contains the text "continue with the review". Below the editor, there is a note: "Attach files by dragging & dropping, selecting them, or pasting from the clipboard." At the bottom of the comment box are buttons for "Cancel" and "Add review comment".

At the bottom of the interface, there are two buttons: "Start a new conversation" and "Finish your review".



# Finish the review

The screenshot shows a GitHub pull request review interface. At the top, the browser address bar displays the URL: <https://github.com/BoiseState/CS481-Test/pull/20/commits/04d51ef8cd13e552a58144e78128ca33857243e3>. The page header indicates "Changes from all commits" with a dropdown, "1 file", and a status of "+5 -0". Buttons for "Unified", "Split", and "Review changes 2" are visible. A commit message "bgdndit committed 14 minutes ago" and the commit hash "04d51ef8cd13e552a58144e78128ca33857243e3" are shown. The main content area displays a code diff for "src/Main.java". The diff shows a new method "demoMethodTask19()" being added, highlighted in green. Below the diff, a conversation thread shows two messages from "bgdndit": "leave feedback directly in the code..." (29 seconds ago) and "continue with the review" (just now), both marked as "Pending". At the bottom, buttons for "Start a new conversation" and "Finish your review" are present. The code diff also shows the "demoMethodIllustrateCommitLink()" method.

implementation of task # x

GitHub, Inc. [US] | <https://github.com/BoiseState/CS481-Test/pull/20/commits/04d51ef8cd13e552a58144e78128ca33857243e3>

Changes from all commits ▾ 1 file ▾ +5 -0 ■■■■■

Unified Split Review changes 2 ▾

bgdndit committed 14 minutes ago commit 04d51ef8cd13e552a58144e78128ca33857243e3

5 ■■■■■ src/Main.java Show comments View

```
... @@ -1,5 +1,10 @@
1 public class Main
2 {
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
```

1 public class Main  
2 {  
3 + private void demoMethodTask19()  
4 + {  
5 + //implementation of task #19 in testPRBranch  
6 + }  
7 +  
8 private void demoMethodIllustrateCommitLink()  
9 {  
10 //initial implementation of

bgdndit 29 seconds ago Pending  
leave feedback directly in the code...

bgdndit just now Pending  
continue with the review

Reply...

Start a new conversation Finish your review

6 + }  
7 +  
8 private void demoMethodIllustrateCommitLink()  
9 {  
10 //initial implementation of

# Finish the review and provide summary

The screenshot shows a GitHub pull request for the repository `BoiseState / CS481-Test`. The pull request is titled `implementation of task #19 #20` and is from the `testPRBranch` to the `master` branch. The reviewer, `bgdndit`, has submitted 2 pending comments. A modal is open for submitting the review, with the following options:

- ☒ **Comment**  
Submit general feedback without explicit approval.
- ☐ **Approve**  
Submit feedback and approve merging these changes.
- ☐ **Request changes**  
Submit feedback that must be addressed before merging.

The modal also includes a text input for the review summary, which currently contains the text `summary of review...`, and buttons for `Cancel review` and `Submit review`.

The background shows the pull request details, including the file `src/Main.java` with changes from all commits. The changes are summarized as `1 file` with `+5 -0` changes. The code snippet for `src/Main.java` is as follows:

```
5 src/Main.java
... @@ -1,5 +1,10 @@
1 public class Main
2 {
3 + private
4 + {
5 + //im
```

# The review session inside the PR #20

The screenshot shows a GitHub pull request (PR) titled "implementation of task # ..." with the URL <https://github.com/BoiseState/CS481-Test/pull/20#pullrequestreview-16052664>. The review is by user "bgdndit".

**Review Comments:**

- bgdndit commented 5 minutes ago: leave feedback for the change
- bgdndit reviewed just now: summary of review...

**Code Diff:**

```
src/Main.java
... @@ -1,5 +1,10 @@
1   1   public class Main
2   2   {
3   +   private void demoMethodTask19()
4   +   {
5   +       //implementation of task #19 in testPRBranch
```

**Review History:**

- bgdndit just now: leave feedback directly in the code...
- bgdndit just now: continue with the review

**Right Sidebar:**

- Milestone: None yet
- Sprint: Sprint 1
- Estimate: No estimate yet
- Epics: Not inside an Epic
- Notifications: Unsubscribe (You're receiving notifications because you were assigned.)
- 1 participant
- Lock conversation
- Move Issue

**Bottom Status:**

Add more commits by pushing to the **testPRBranch** branch on **BoiseState/CS481-Test**.

**Review requested** (Green checkmark icon)  
Review has been requested on this pull request. It is not required to merge. [Learn more.](#) [Show all reviewers](#)

# Address the changes for task #19 based on the feedback from the PR review session

The screenshot shows a code editor interface with a sidebar on the left and a main editor area. The sidebar contains a search bar, a repository list with 'CS481-Test' selected, and a 'Tutorial' link. The main editor area shows a diff for 'src\Main.java' in the 'testPRBranch'. The diff highlights changes for task #19, specifically adding comments and code for 'demoMethodTask19()'. The changes are summarized in a table below the diff.

File	Line	Change
src\Main.java	3	@@ -3,6 +3,8 @@ public class Main
src\Main.java	4	private void demoMethodTask19()
src\Main.java	5	{
src\Main.java	6	//implementation of task #19 in testPRBranch
src\Main.java	7	//addressed some changes for task #19
src\Main.java	8	//based on the feedback from the PR review session
src\Main.java	9	}
src\Main.java	10	private void demoMethodIllustrateCommitLink()

Summary

Description

Commit to testPRBranch

# Add a new commit to the testPRBranch

Filter repositories

GitHub

CS481-Test

Other

Tutorial

testPRBranch

Changes History

#20

Update from master View branch Sync

1 change

src/Main.java

...	...	@@ -3,6 +3,8 @@
3	3	private void demoMethodTask19()
4	4	{
5	5	//implementation of task #19 in testPRBranch
6	6	//addressed some changes for task #19
7	7	//based on the feedback from the PR review session
6	8	}
7	9	
8	10	private void demoMethodIllustrateCommitLink()

updated task #19

Description

Commit to testPRBranch

# The new commit changes appear in PR #20

The screenshot shows a web browser window displaying a GitHub Pull Request (PR) #20. The browser's address bar shows the URL: <https://github.com/BoiseState/CS481-Test/pull/20#pullrequestreview-16052664>. The page title is "implementation of task #".

The main content area shows a review by user **bgdndit** 4 minutes ago. The review includes a summary of review... and a code diff for `src/Main.java`. The code changes are highlighted in green, showing the addition of a `demoMethodTask19()` method.

The code diff shows the following changes:

```
... @@ -1,5 +1,10 @@
1 1 public class Main
2 2 {
3 + private void demoMethodTask19()
4 + {
5 + //implementation of task #19 in testPRBranch
```

Below the code diff, there are two comments from **bgdndit** 4 minutes ago:

- leave feedback directly in the code...
- continue with the review

A red dashed box highlights a comment from **bgdndit** 4 minutes ago: "updated task #19". The comment is associated with the commit hash `08644da`.

At the bottom of the page, a green box indicates a "Review requested" status. The message states: "Review has been requested on this pull request. It is not required to merge. [Learn more.](#)"

The right sidebar contains the following sections:

- Sprint 1**
- Estimate**: No estimate yet
- Epics**: Not inside an Epic
- Notifications**: Unsubscribe button. Message: "You're receiving notifications because you were assigned."
- 1 participant**: bgdndit
- Lock conversation**
- Move Issue**

# Merge the pull request into the master branch

The screenshot shows a GitHub pull request page for the repository `BoiseState/CS481-Test`. The pull request is titled `implementation of task #` and is currently in the `testPRBranch` state. The page displays a conversation with two messages from user `bgdndit`, both posted 6 minutes ago. The first message says "leave feedback directly in the code..." and the second says "continue with the review". There is a "Reply..." input field below the messages. On the right side, there is a "1 participant" section with a lock icon and a "Lock conversation" button, and a "Move Issue" button.

Below the conversation, there is a status bar indicating that the pull request was updated to `task #19` with commit `08644da`. A notification from `bgdndit` states "removed the request for review from shanep 13 seconds ago".

The main content area shows a green checkmark icon and the message: "This branch has no conflicts with the base branch. Merging can be performed automatically." Below this message is a green button labeled "Merge pull request" with a dropdown arrow. To the right of the button, it says "You can also open this in GitHub Desktop or view command line instructions." A dropdown menu is open, showing three options:

- Create a merge commit**: All commits from this branch will be added to the base branch via a merge commit.
- Squash and merge**: The 2 commits from this branch will be combined into one commit in the base branch.
- Rebase and merge**: The 2 commits from this branch will be rebased and added to the base branch.

At the bottom of the page, there is a text input field for a commit message, with a placeholder text "or pasting from the clipboard." and a "Commit" button.

# Finish the PR by merging it into the master branch

The screenshot shows a web browser window with the GitHub interface. The address bar displays the URL: <https://github.com/BoiseState/CS481-Test/pull/20#pullrequestreview-16052664>. The page title is "implementation of task #19".

At the top, a comment from user **bgdndit** (8 minutes ago) says "continue with the review". Below it is a "Reply..." input field. To the right, there are buttons for "Lock conversation" and "Move Issue".

A commit history section shows "updated task #19" with commit hash `08644da`. Below that, a message indicates that **bgdndit** removed the request for review from **shanep** 2 minutes ago.

A message prompts the user to "Add more commits by pushing to the `testPRBranch` branch on `BoiseState/CS481-Test`."

The main section shows a merge dialog. It contains a text box with "Merge pull request #20 from BoiseState/testPRBranch" and another text box with "implementation of task #19". Below these are two buttons: "Confirm merge" (green) and "Cancel".

Below the merge dialog is a comment section. It has a "Write" tab and a "Preview" tab. The "Write" tab is active, showing a rich text editor with a toolbar (bold, italic, quote, code, link, list, etc.). The text area contains the placeholder "Leave a comment". Below the text area is a note: "Attach files by dragging & dropping, selecting them, or pasting from the clipboard." At the bottom of the comment section are two buttons: "Close pull request" and "Comment" (green).

At the very bottom, a small note says "Styling with Markdown is supported".



# After successful PR merge and close, delete the branch

The screenshot shows a web browser window with the GitHub interface. The address bar displays the URL: <https://github.com/BoiseState/CS481-Test/pull/20#pullrequestreview-16052664>. The page title is "implementation of task #".

The pull request history shows the following actions:

- updated task #19 (commit 08644da)
- bgdndit removed the request for review from shanep 2 minutes ago
- bgdndit merged commit 5a16620 into master just now (with a [Revert](#) button)

A notification banner states: "Avoid bugs by automatically running your tests. Continuous integration can help catch bugs by running your tests automatically. Merge your code with confidence using one of our continuous integration providers." (with a [Learn more](#) button).

The main content area shows a message: "Pull request successfully merged and closed. You're all set—the testPRBranch branch can be safely deleted." (with a [Delete branch](#) button).

Below this is a comment section with a "Write" tab selected. The text area contains the placeholder "Leave a comment" and a note: "Attach files by dragging & dropping, selecting them, or pasting from the clipboard." (with a [Comment](#) button).

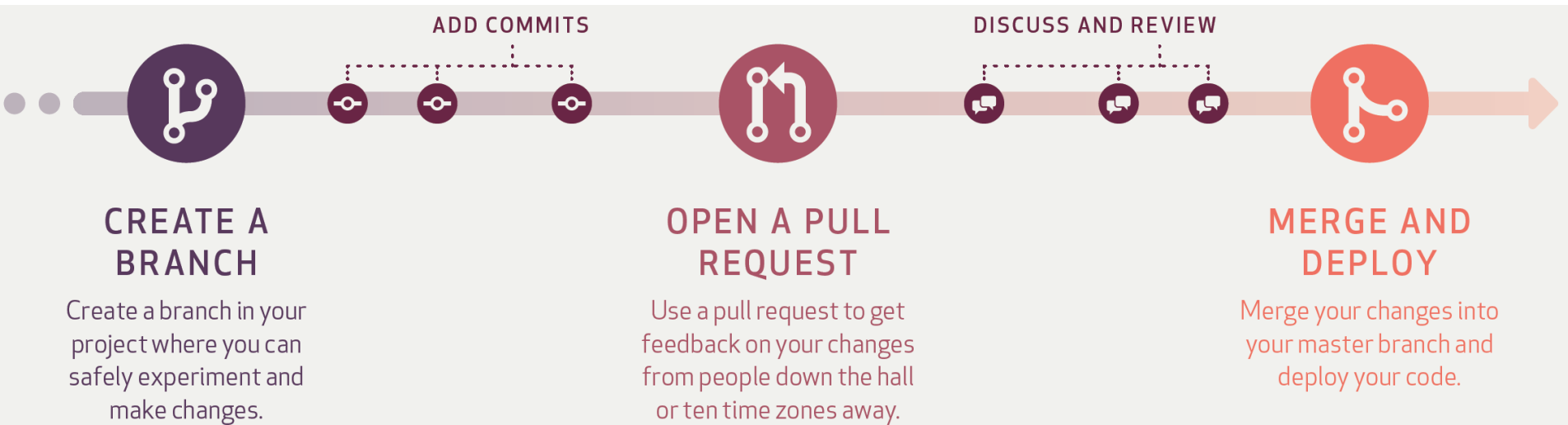
A "ProTip!" message at the bottom says: "Add comments to specific lines under [Files changed](#)."

The footer contains copyright information: "© 2017 GitHub, Inc." and links to [Terms](#), [Privacy](#), [Security](#), [Status](#), [Help](#), [Contact GitHub](#), [API](#), [Training](#), [Shop](#), [Blog](#), and [About](#).

# GitHub Workflow

# GitHub Workflow

- <https://guides.github.com/introduction/flow/>



- Best suited for open-source software development