

# **Работа с репозиторием Mercurial/Git в BitBucket**

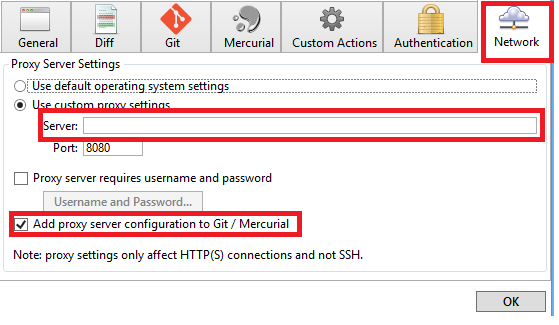
**Bitbucket** («ведро битов») — веб-сервис для хостинга проектов и их совместной разработки, основанный на системе контроля версий Mercurial и Git. В данной инструкции используется именно он.

## Клиентское ПО

Для начала работы с репозиториями необходимо установить клиент **Source Tree** скачав его с официального сайта<https://www.sourcetreeapp.com/>

После установки клиента возможно понадобится настройка подключения к сети

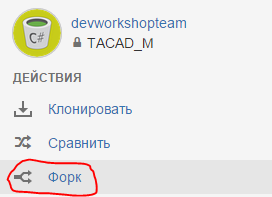
Server: proxy.isd.dp.ua



## Форк

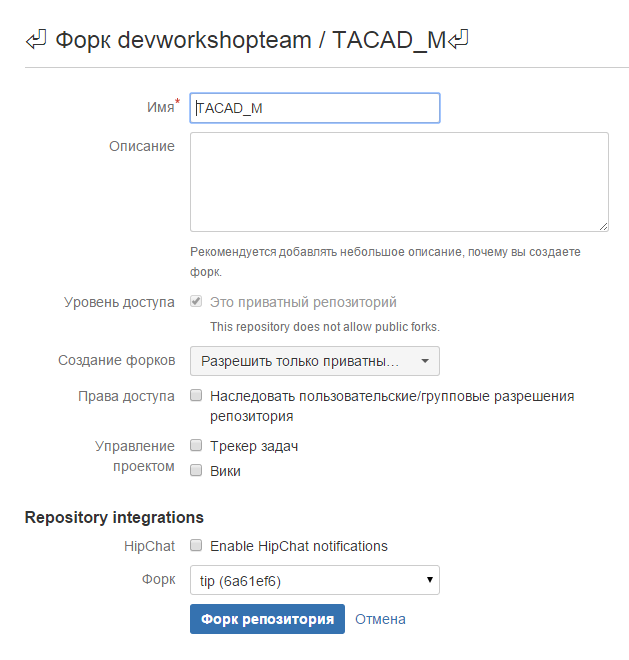
Форк, (англ. fork — ответвление) — использование кодовой базы программного проекта в качестве старта для другого. При этом основной проект может как продолжать существование, так и прекратить его. Синонимы: ответвление, отпочковывание. Ответвленный проект может поддерживать и обмениваться частью содержимого с основным проектом, а может и приобрести абсолютно другие свойства, перестав иметь с базовым проектом что-то общее.

### Создание форка



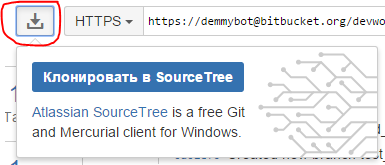
Что бы создать собственный форк необходимо открыть репозиторий-донор и нажать соответствующую кнопку.

В появившемся окне необходимо задать Ваше название проекта и есть возможность задать дополнительные опции.

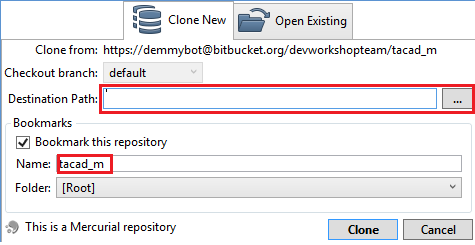


после этого в вашем аккаунте появится полностью идентичная копия репозиторя-донора с которой можно работать.

### Клонирование на локальную машину

Созданный на предыдущем шаге репозиторий физически размещен на серверах Bitbucket и что бы начать с ним работу нужно получить его локальную(рабочую) копию.

Самым простым решением данного вопроса при установленном Source Tree является опция клонирования.

Для этого откройте на Bitbucket **свой** репозиторий и нажмите кнопку клонировать в Source Tree в результате этого у вас откроется соответствующее диалоговое окно 

В **Destination path** необходимо указать папку на локальном диске куда будет клонирован ваш репозиторий.

По окончании настроек нажмите кнопку **Clone**

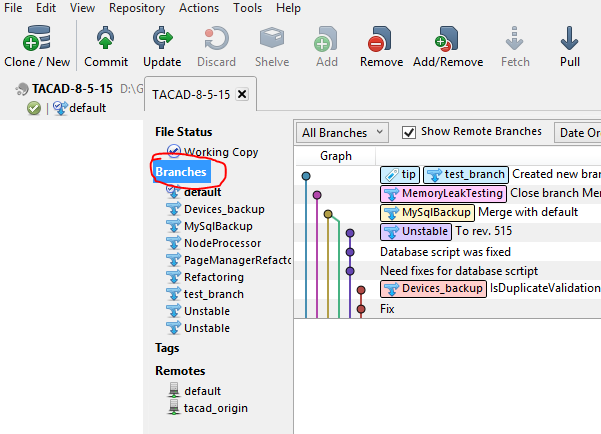
## Работа над задачами

### Бренч

Для работы над определенной задачей необходимо создать специальное, присущее этой, конкретной, задаче ответвление кода. Такие ветвления именуются ветками (branch) и создаются отдельно под каждую задачу.

При создании ветки в нее копируется текущее состояние кода по этому перед созданием **настоятельно рекомендуется провести обновление из мастер-репозитория** (см. Подключение мастер-репозитория)

Для создания новой ветки в Source Tree необходимо кликнуть **правой** кнопкой мыши на вкладке **Branches** и выбрать пункт **New Branch**

****

В появившемся окне необходимо ввести название ветки и нажать кнопку **Create branch**

**ВНИМАНИЕ!: название ветки должно быть значимым и отражать суть задачи для которой эта ветка формируется.**

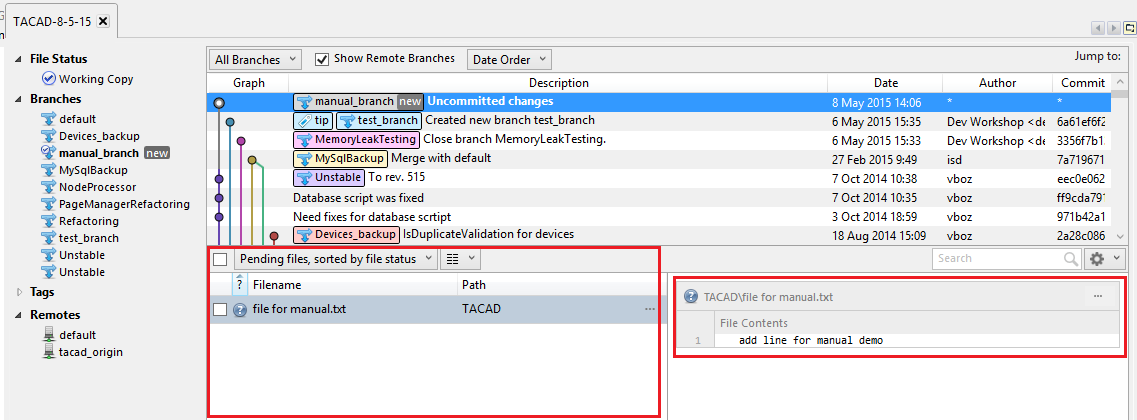
После этого созданная ветка появится в списке **Branches** и на нее можно переключиться двойным кликом мыши.

Активная ветка помечена специальный значком и выделена жирным шрифтом.

### Фиксация изменений

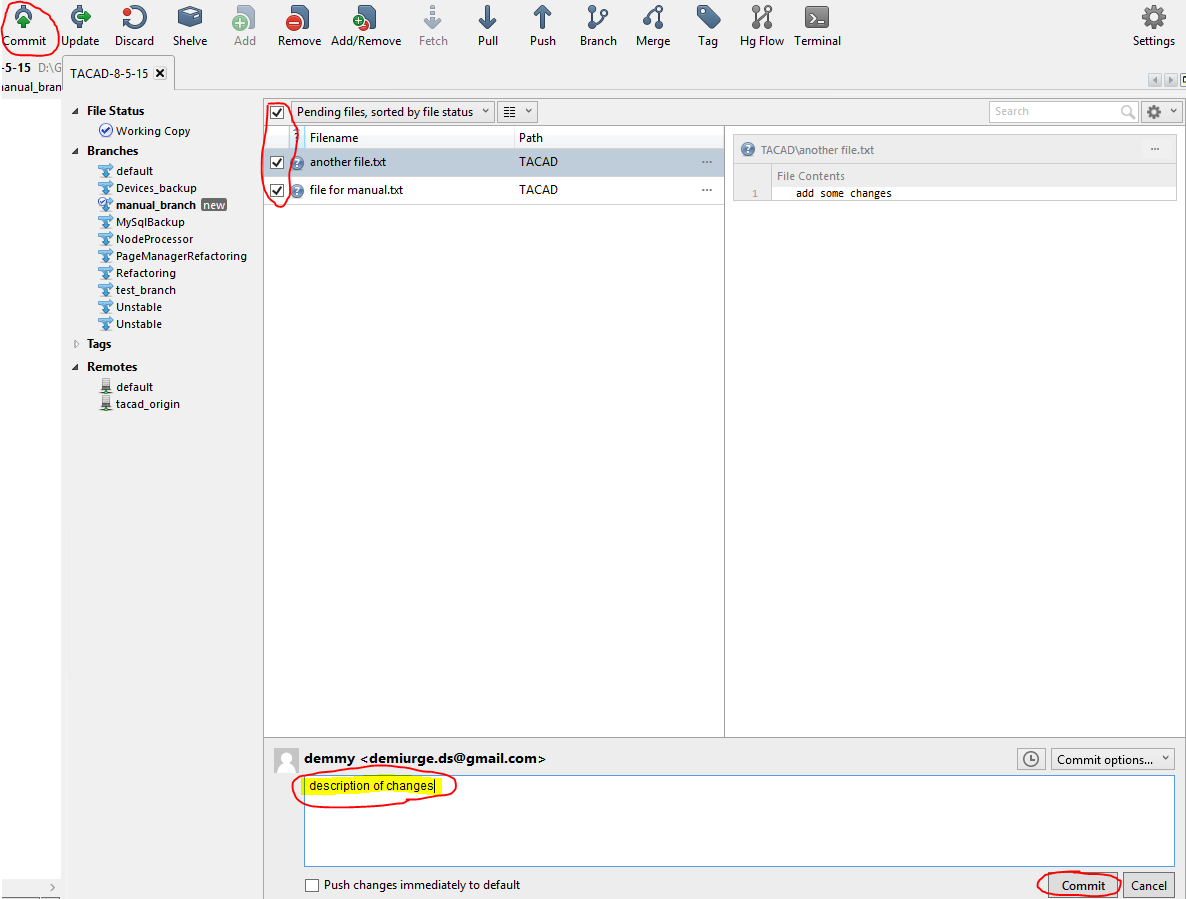
После внесения изменений в программу их необходимо зафиксировать. Фиксация внесенных изменений называется **комИт**(commit) и записывает разницу состояний до и после изменений.

По мере внесения изменений в программу они фиксируются Source Tree и вы всегда можете отследить какие файлы подвергались изменению и какие строки в них менялись



Что бы зафиксировать изменения в файлах необходимо:

* нажать кнопку **Commit** в верхнем меню.
* в появившемся окне выделить файлы которые вы хотите зафиксировать
* ввести описание внесенных изменений
* нажать кнопку Commit внизу формы



**ВНИМАНИЕ!: описание внесенных изменеий должно быть значимым и отражать суть решения которое фиксируется.**

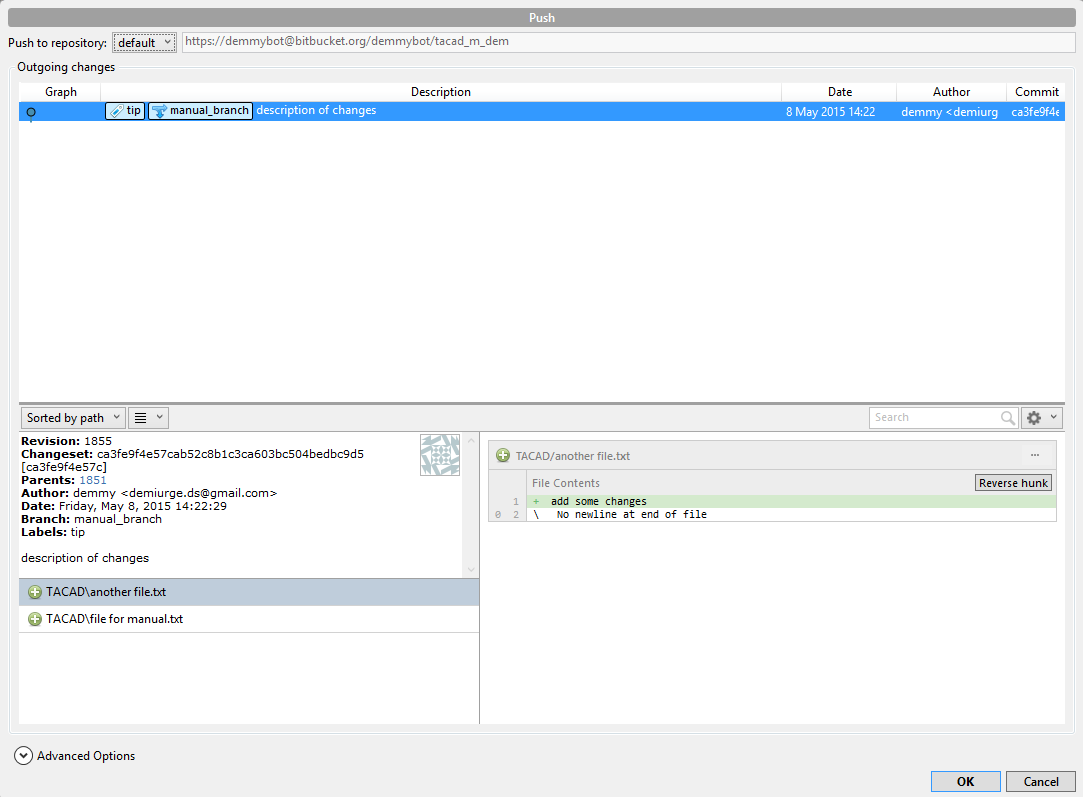
### Загрузка изменений на сервер

Что бы сохранить изменения в вашем репозитории их необходимо загрузить на сервер командой **Push** основного меню.

В появившемся окне вы можете проверить:

* репозиторий в который вы отправляете данные
* список отправляемый коммитов
* изменные файли
* изменения по каждому файлу

После проверки необходимо нажать **Ok** и данные будут отправленны на сервер.



*Если текущая ветка на сервере отсутствует она будет создана автоматически.*

## Слияние веток

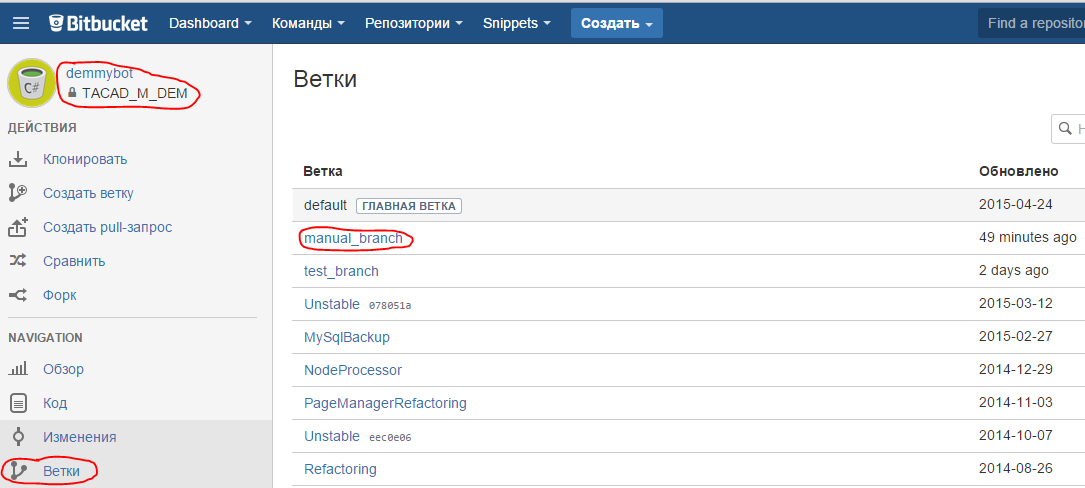
По окончании решения рабочей задачи весь пакет изменений необходимо направить на слияние с основным кодом.

Для этого необходимо сформировать пул-реквест(pull-reqest) непосредственно в **Bitbucket**

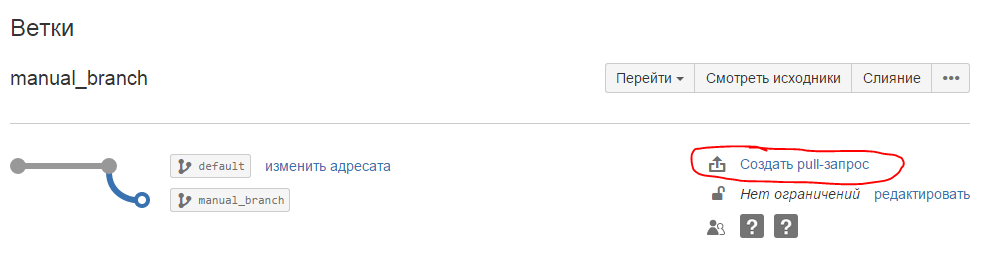
Перед формированием пул реквеста необходимо получить последние изменения из мастер-репозитория (см. Подключение мастер-репозитория)

### Создание пул-реквеста

Для создания запроса на слияние необходимо в **своем** репозитории **Bitbucket** выбрать ветку в которой вы работали



В выбранной ветке Создать pull-запрос

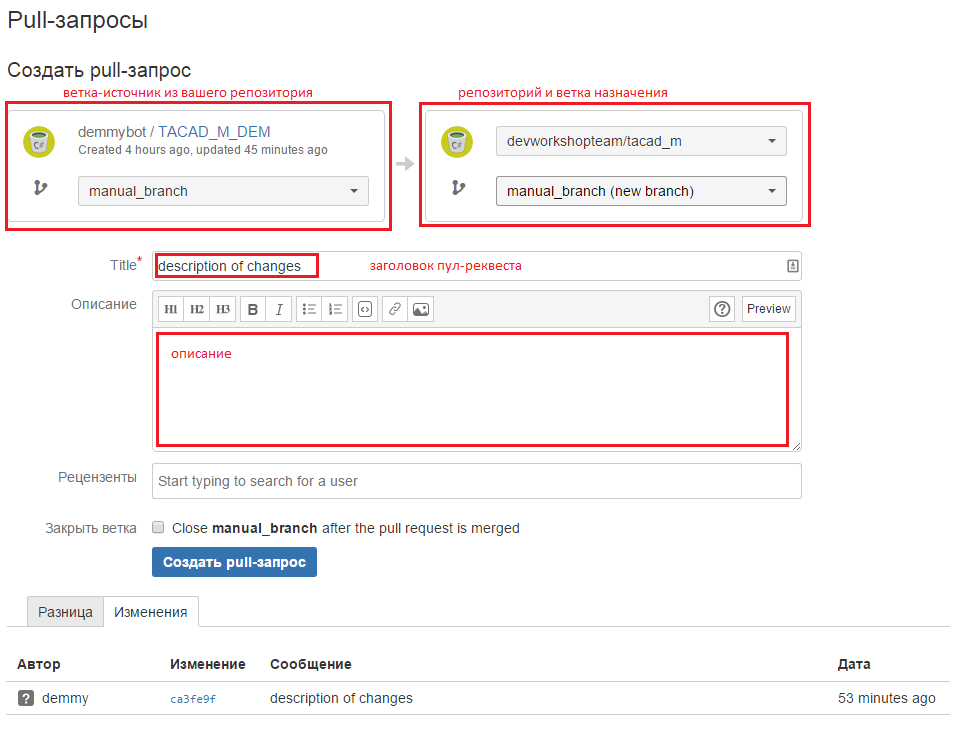


В окне создания pull-запроса необходимо выбрать:

1. Ветку-источник с изменениями

2. Репозиторий назначения

3. Ветку в репозитории назначения



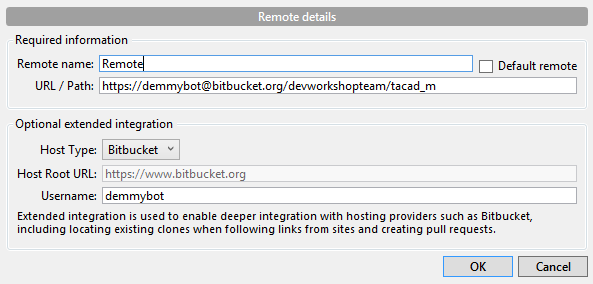
## Подключение мастер-репозитория

Поскольку параллельно с Вами работу над проектом ведут другие люди существует необходимость периодически синхронизировать изменения вносимые ими в основной репозиторий.

Для этого его нужно подключить как дополнительный (для чтения) внешний репозиторий.

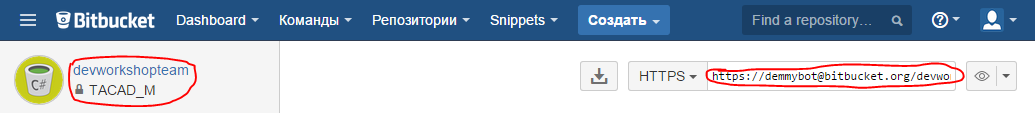
Для этого необходимо кликнуть правой кнопкой мыши на вкладке **Remotes** и выбрать пункт **New Remote...**

В появившемся окне необходимо ввести название для мастер-репозитория и его URL (мастер - репозиторий **с которого** был сделан форк)



Что бы узнать URL мастер репозиторий можно зайти в него на **Bitbucket** и скопировать полный путь.

**ВНИМАНИЕ!: убедитесь, что именно в мастер репозитории С КОТОРОГО вы делали форк, а не в своем форке.**

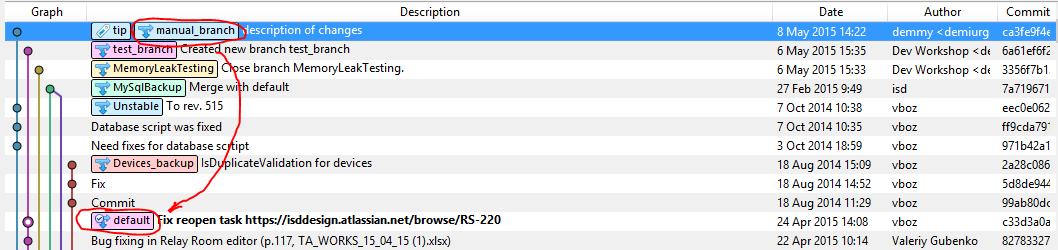
****

### Синхронизация с мастер-репозиторием

1. Обновите мастер-репозиторий кликнув на нем правой кнопкой мыши и выбрав пункт **Pull from <ИМЯ\_МАСТЕРА>**
2. Синхронизируйте данные нажав кнопку **Merge** основного меню и выбрав в списке комитов ваш мастер-репозиторий

**Например: вненсение изменений из manual\_branch в default**

Активная ветка **default** выделена специфическим значком и жирным шрифтом. Ветка **manual\_branch** из которой берутся изменение выделена кликом мыши подсвечена синим. При нажатии **Ok** изменения внесенные в manual\_branch попадут в default



## Разрешение конфликтов

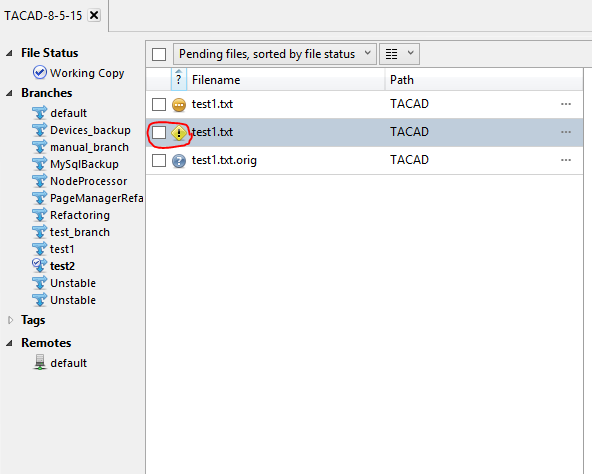
Для разрешения конфликтов в визуальном режиме необходимо установить и подключить **merge tool**. Source tree поддерживает работу со следующими программами:

* Araxis merge
* Beyond compare
* DiffMerge
* KDiff3
* K4Merge
* TortoiseMerge

Мы используем KDiff3(GPL) <http://sourceforge.net/projects/kdiff3/files/> в связи с его доступностью и функционалом

Для подключения KDiff3 нужно зайти в верхнем меню в Раздел **Tools->Opions** и в разделе **Diff->Merge tool** выбрать **KDiff3** из выпадающего списка

Конфликтующие фалы имеют различные пометки. Кликните правой кнопкой по файлу помеченному вопросительным знаком и выберите**Resolve conflicts->Launch External Merge Tool**



**Альтернативный клиент**

Soure Tree, по офиициальной документации, требует ОС не ниже Windows 7. При работе на более старых ОС вы можете воспользоваться **Tortoise Hg** <http://tortoisehg.bitbucket.org/> возможности и принципы работы которого аналогичны с Soure Tree