

Шпаргалка svn

Наполнение репозитория Subversion

Репозиторий, созданный командой:

```
$ svnadmin create /usr/local/svn/newrepos
```

нужно наполнить свежими данными. Для этого есть команда *svn import* это быстрый способ скопировать не версионированное дерево файлов в хранилище, например:

```
$ svn import mytree file:///usr/local/svn/newrepos/some/project \
```

```
-m "Initial import"
```

```
Adding mytree/foo.c
```

```
Adding mytree/bar.c
```

```
Adding mytree/subdir
```

```
Adding mytree/subdir/quux.h
```

```
Committed revision 1.
```

Стоит заметить, что после завершения импорта, оригинальное дерево файлов не конвертируется в рабочую копию. Для того, чтобы начать работать вам необходимо создать новую рабочую копию (*svn checkout*) дерева файлов.

Добавление новых файлов и каталогов в репозиторий Subversion

Например, вы создаёте крупный документ в ЛaTeXe, и появился новый каталог с новыми текстами или картинками. Как их добавить в Subversion репозиторий? Очень просто:

```
$ svn add temp A temp/Lit_obzor2_edt.tex
```

Теперь смотрим на статус наших файлов и видим, что против них стоит буква **A** - значит, они **Added**, то есть добавлены:

```
$ svn status
```

```
? image
```

```
? biblio
```

```
? optcodeimages
```

```
M draft_PhD_Konnik.dvi
```

```
! PhD_Konnik.log
```

```
A temp
```

```
A temp/Lit_obzor2_edt.tex
```

```
! PhD_Konnik.toc
```

```
M PhD_Konnik_chapter_results.tex~
! PhD_Konnik_chapter_introduction.dvi
! PhD_Konnik.aux
! PhD_Konnik.bbl
! draft_PhD_Konnik.log
! PhD_Konnik.blg
! draft_PhD_Konnik.toc
! draft_PhD_Konnik.aux
! draft_PhD_Konnik.bbl
M PhD_Konnik.dvi
! draft_PhD_Konnik.blg
M PhD_Konnik_chapter_results.tex
! PhD_Konnik_chapter_introduction.log
```

Они будут помещены в репозиторий при следующей отправке (commit).

Обновление рабочей копии до состояния репозитория

Файлы, находящиеся в репозитории свежее, чем у вас в рабочей копии. Дубовый вариант заключается в том, чтобы удалить свою рабочую копию и сделать checkout из репозитория. Это проходит, если репозиторий только ваш и вы уверены, что никаких изменений в рабочей копии вы не потеряете. Но есть вариант и лучше:

```
$ svn update U
```

```
biblio/my.bib U
```

```
biblio/my.bib.bak
```

```
Updated to revision 9.
```

Это обновит вашу локальную копию до ревизии, которая находится в репозитории SVN. После этого можно приступать к работе с файлами, так как они обновлены до последней версии.

Латинские буквы перед именем файла имеют свой сакральный смысл, скрытый в недрах документации сокрушительных размеров (кстати, документация очень хороша, и есть даже частичный перевод на русский [тут](#)). Собственно, буква означает действие, которое выполнил Subversion для приведения рабочей копии в актуальное состояние:

```
U foo
```

Updated — Файл foo был обновлен

A foo

Added — Файл или директория foo были добавлены в рабочую копию.

D foo

Deleted — Файл или директория foo были удалены из рабочей копии.

R foo

Replaced — Файл или директория foo была заменена в рабочей копии; это значит, что foo был удален, а новый элемент с таким же именем был добавлен.

G foo

merGed — Файл foo получил новые изменения из хранилища, однако ваша локальная копия содержит ваши изменения. Либо изменения не пересекаются, либо они точно такие же, как ваши локальные изменения, поэтому Subversion успешно выполнил слияние изменений хранилища с файлом.

C foo

Conflicting — Файл foo получил от сервера конфликтующие изменения. Изменения с сервера пересекаются с вашими изменениями фала.

Просмотр истории ревизий в Subversion

Часто нужно посмотреть, что и как редактировалось - для этого при отправке новых данных в репозиторий (commit) есть смысл писать содержательные и осмысленные комментарии. Если это так, то можно многое узнать о том, что и когда редактировалось:

```
$ svn log --revision 1:3
```

что значит: выдать лог ревизий с первую по третью. Увидим:

```
----- r1 | penta4 | 2008-06-03 15:49:21 +0400 (Втр, 03 Июнь 2008) | 1
line Initial dissertation state
----- r2 | penta4 | 2008-06-03 16:56:01 +0400 (Втр, 03 Июнь 2008) | 3
lines Committed new editions
----- r3 | penta4 | 2008-06-03 16:58:12 +0400 (Втр, 03 Июнь 2008) | 2
lines incremental -----
```

Очень удобно, когда нужно кому-то отправить историю правок вашей программы или документа.

Просмотр изменений в Subversion

Например, нужно просмотреть изменения, произошедшие со второй по четвёртую ревизии. Легко!

```
$ svn diff --revision 2:4 PhD_Konnik.tex > PhD_Konnik.tex.diff.r2-4
```

Это выведет нам в diff-файл все изменения, а ваш любимый текстовый редактор их с удовольствием подсветит для более удобного просмотра.

Отмена изменений в Subversion

Вы сидите и правите какой-то документ - и тут с ужасом понимаете, что правите не тот документ. Не беда - переносите изменения во временный файл, а для того документа, который вы нечаянно испортили, пишем:

```
$ svn revert UMNIK08-Report-01/Report-01-sub1-shortened.tex
```

```
Reverted 'UMNIK08-Report-01/Report-01-sub1-shortened.tex'
```

После этого SVN вернёт файл *Report-01-sub1-shortened.tex* в то состояние, в котором он находится в репозитории.

Копирование или перемещение нескольких файлов в SVN/Subversion (multiple targets move/copy in SVN)

Странно, однако SVN не понимает символов типа *, так что ваша попытка сделать так:

```
$ svn mv MCBC/*jpg LinuxImages/
```

приведёт к ошибке:

```
svn: Client error in parsing arguments
```

Чтобы не перемещать все файлы по одному, воспользуемся ваша любимая оболочка тут, в моём случае это Bash:

```
for i in *.cs ; do svn mv $i newFolder/ ; done
```

Конечно, если в вашей операционной системе есть нормальная консоль :-)

Выдача определённой правки файла из репозитория

Например, нам очень нужно посмотреть, каким был файл при второй правке. Для этого пишем:

```
$ svn cat --revision 2 PhD_Konnik.tex > PhD_Konnik.tex.v2
```

То есть выводим результат второй правки файла *PhD_Konnik.tex* и записываем его в файл *PhD_Konnik.tex.v2* для дальнейшего просмотра.

Восстановление удалённых файлов в Subversion

Ну и наконец самое страшное: вы приказали Subversion удалить файл из репозитория, а потом кто-то вам говорит: этот файл нужно вернуть. На самом деле, файл, который вы удалили из SVN, никуда не делся: элемент исчезает из правки HEAD, но продолжает существовать в более ранних правках. Так что вопрос "как восстановить удалённый файл в Subversion" имеет довольно простой ответ: скопировать его из последней правки, где он был.

Главное вспомнить, как этот файл назывался. Если помним хотя бы частично имя файла (например, *overview*), то можно попросить Subversion вывести все упоминания о правках этого файла так:

```
svn log -v | grep spatialcoh -A 5 -B 5
```

Здесь параметры для **grep** такие: **-A 5** это выдача 5 строк после (**A**fter) совпадения, **-B 5** это 5 строк до (**B**efore) совпадения. Получим длинный вывод вида:

```
spatialcoherencyexplained.eps 436 | penta4 | 2009-12-02 15:39:30 +0300 (Срд, 02 Дек 2009) |
```

```
2 lines
```

```
Changed paths:
```

```
M /PhD_Konnik_autoreferat.tex
```

```
M /PhD_Konnik_chapter_opticalcoding.tex
```

```
M /PhD_Konnik_chapter_overview.tex
```

```
D /images-opticalcoding/spatialcoherencyexplained.eps
```

```
M /images-opticalcoding/sxemacorrelator3.eps
```

```
M /latex.py
```

```
M /svnstatus.tex
```

```
M /zagashnikPhD_Konnik.tex
```

```
--
```

```
M /biblio
```

```
A /formaldocs/disserDocs.tif.7z
```

```
A /images-opticalcoding/PSFdotssize.eps
```

```
M /images-opticalcoding/PSFoutofLambdaDeltaLambdavsfromNumberOfElementsPSFRus.ps
```

```
A /images-opticalcoding/auxiliaryImagesForDisser.svg
```

A /images-opticalcoding/spatialcoherencyexplained.eps

M /zagashnikPhD_Konnik.tex

TENGIBLE changes:

- autoreferat is improved slightly;

- Spatial coherency is improved

Красным подчеркнул я: видно, что файл **spatialcoherencyexplained.eps** удалён в правку 436, и теперь хотим его восстановить. Смотрим, в какой правке файл был удалён - в 436, значит, в 435 есть его последняя копия. Восстанавливаем:

```
svn copy --revision 435 file:///phdthesis/images-opticalcoding/spatialcoherencyexplained.eps  
spatialcoherencyexplained.eps
```

Всё, теперь файл появится снова, вместе с историей правок (что важно) и вы можете его править дальше.