



VVDN
TECHNOLOGIES

SVN

Subversion Control System.

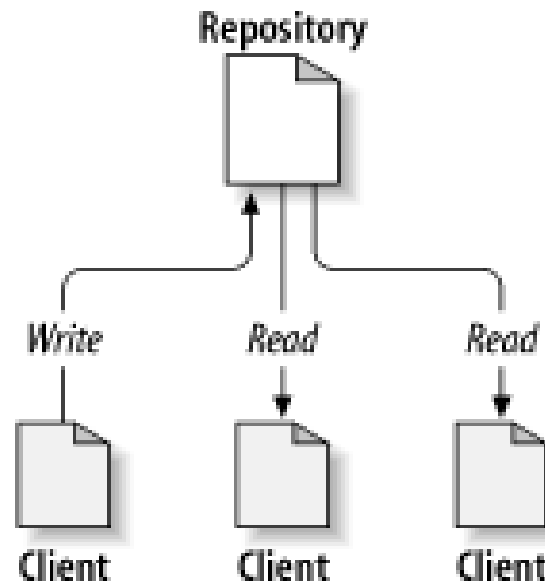
Author:
Aashish Patel
Rakesh Pillai

Basics

- What Version Control system does?
 - Tracks incremental versions (or revisions) of files/directories over time.
- What makes a version control system useful ?
 - It allows you to explore the changes which resulted in each of those versions .
 - facilitates the arbitrary recall of the same.

The Repository

- The core of the version control system.
- The central store of that system's data.
- Stores information in the form of a *filesystem tree*.
- *Any number of clients connect to the repository, and read or write to these files.*

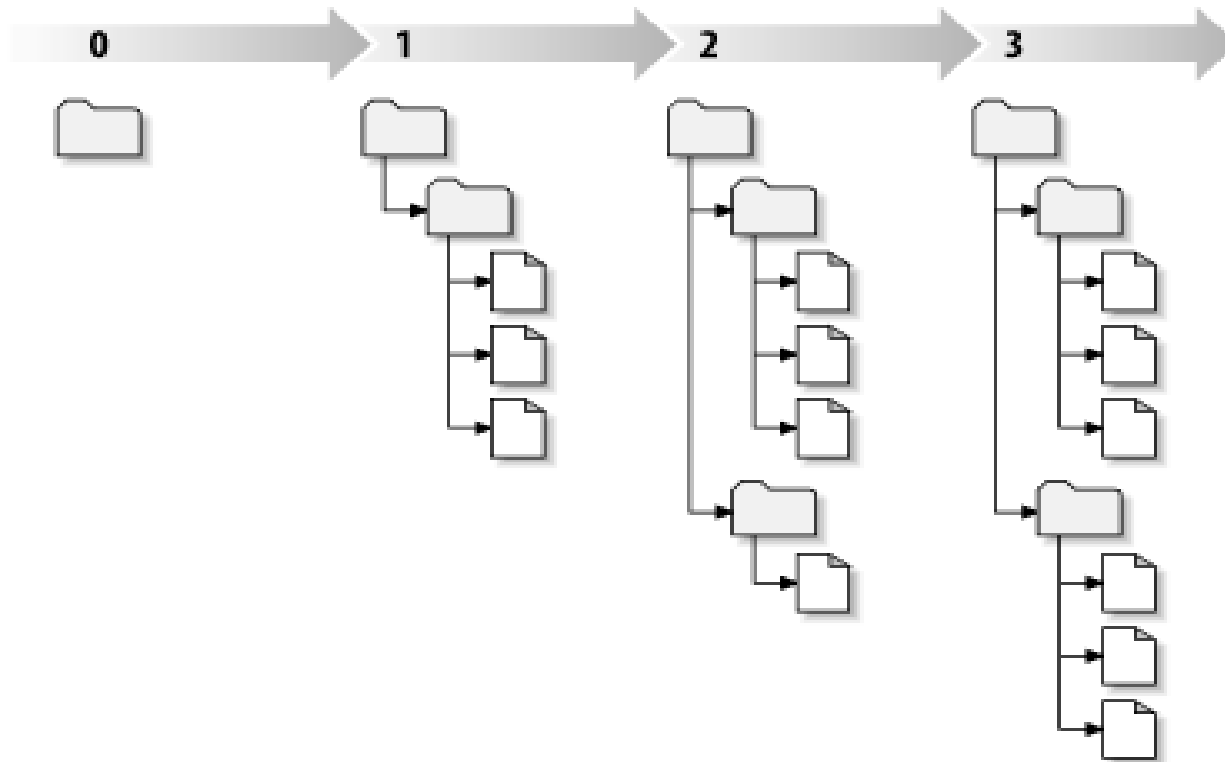


Revisions

- Each commit to repository creates a new state of the filesystem tree, called a *revision*.
- A client commits any no. of files and directories as a single **atomic transaction**.
- That is either all changes are accepted into the repository, or none of them is.

Revisions

- Each revision is assigned a unique natural number, one greater than the number assigned to previous revision.

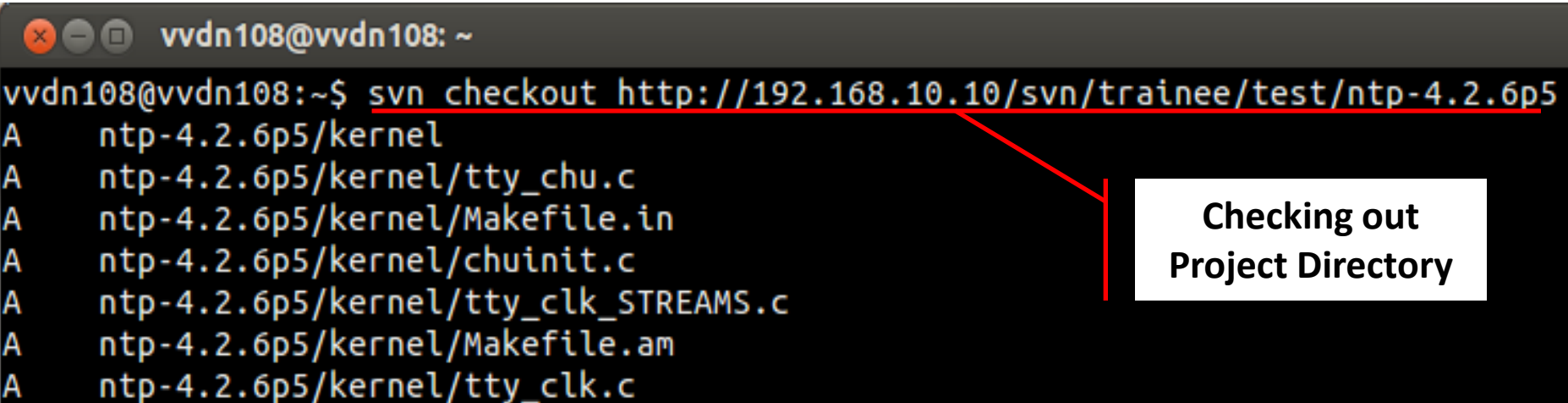


Subversion Working Copies

- It (WC) is an ordinary directory tree on your local system, containing a collection of files.
- You can edit , make changes, compile(if source file) in usual way.
- You can even have multiple working copies of the same project.

Fundamental Working Copy Interactions

- How to get a Working Copy ?
 - To get a working copy, you must ***check out*** *some subtree of the repository.*



```
vvdn108@vvdn108: ~  
vvdn108@vvdn108:~$ svn checkout http://192.168.10.10/svn/trainee/test/ntp-4.2.6p5  
A ntp-4.2.6p5/kernel  
A ntp-4.2.6p5/kernel/tty_chu.c  
A ntp-4.2.6p5/kernel/Makefile.in  
A ntp-4.2.6p5/kernel/chuinit.c  
A ntp-4.2.6p5/kernel/tty_clk_STREAMS.c  
A ntp-4.2.6p5/kernel/Makefile.am  
A ntp-4.2.6p5/kernel/tty_clk.c
```

Checking out
Project Directory

Importing Files and Directories

- You can get new files into your Subversion repository in two ways: **svn import** and **svn add**.

```
vvdn108@vvdn108: ~/ntp-4.2.6p5
vvdn108@vvdn108:~/ntp-4.2.6p5$ svn add testdir
A      testdir
A      testdir/myfile.c
A      testdir/dir1
A      testdir/dir1/file1
A      testdir/dir2
A      testdir/dir2/file2
vvdn108@vvdn108:~/ntp-4.2.6p5$
vvdn108@vvdn108:~/ntp-4.2.6p5$ svn add testfile
A      testfile
vvdn108@vvdn108:~/ntp-4.2.6p5$
```

Adding Directory

Adding a file

Importing Files and Directories

- **svn import** command used to copy an tree of files into a repository.
- It doesn't require a working copy
- Your files are ***immediately committed*** to the repository.

```
vvdn108@vvdn108: ~/ntp-4.2.6p5
vvdn108@vvdn108:~/ntp-4.2.6p5$ svn import testdir http://192.168.10.10/svn/trainee/ntp-4.2.6p5/
-m "Adding repo to the server."
Adding      testdir/dir1
Adding      testdir/dir1/file1
Adding      testdir/dir2
Adding      testdir/dir2/file2
Adding      testdir/myfile.c

Committed revision 1408.
vvdn108@vvdn108:~/ntp-4.2.6p5$
```

Adding Directory

Directly Committed

Other Useful SVN commands

- **svn copy (cp)** — Copy a file or directory in a WC or in the repository. SRC and DST can be WC/URL .

svn copy SRC[@REV]... DST

```
vvdn108@vvdn108: ~/trainee/testdir
vvdn108@vvdn108:~/trainee/testdir$ svn copy myfile1.c wrk_cpy.c
A      wrk_cpy.c
vvdn108@vvdn108:~/trainee/testdir$ svn copy myfile2.c http://192.168.10.10/svn/trainee/test/foo.c
                                     -m "Adding files to url."
Adding copy of      myfile2.c
Committed revision 1417.
vvdn108@vvdn108:~/trainee/testdir$ svn copy http://192.168.10.10/svn/trainee/test/myfile1.c bar.c
A      bar.c
vvdn108@vvdn108:~/trainee/testdir$ svn copy http://192.168.10.10/svn/trainee/test/myfile1.c
                                     http://192.168.10.10/svn/trainee/test/foo_bar.c -m "Remote copy"
Committed revision 1418.
vvdn108@vvdn108:~/trainee/testdir$
```

WC to WC

WC to URL

URL to URL

Other Useful SVN commands

- **svn list (ls)** — List directory entries in the repository.
- **svn mkdir** — Create a new directory under version control.

```
vvdn108@vvdn108: ~/trainee/testdir/dir1
vvdn108@vvdn108:~/trainee/testdir/dir1$ svn list http://192.168.10.10/svn/trainee/testdir/dir1
dir3/
file1
file3.c
vvdn108@vvdn108:~/trainee/testdir/dir1$ svn mkdir dir4
A      dir4
vvdn108@vvdn108:~/trainee/testdir/dir1$
```

Other Useful SVN commands

- **svn move (mv)** — Move a file or directory.
- Equivalent to an *svn copy* followed by *svn delete*.

vvdn108@vvdn108: ~/trainee/testdir

```
vvdn108@vvdn108:~/trainee/testdir$ svn move myfile1.c wrk_mov.c
```

```
A      wrk_mov.c
```

```
D      myfile1.c
```

WC to WC

```
vvdn108@vvdn108:~/trainee/testdir$ svn move http://192.168.10.10/svn/trainee/test/myfile2.c  
http://192.168.10.10/svn/trainee/test/bar_foo.c -m "Remote move"
```

URL to URL

```
Committed revision 1420.
```

```
vvdn108@vvdn108:~/trainee/testdir$
```

Other Useful SVN Commands

- **svn delete** (del, remove, rm) — Delete an item from a working copy or the repository.

```
vvdn108@vvdn108: ~/trainee/testdir
vvdn108@vvdn108:~/trainee/testdir$ svn delete dir2
D      dir2
D      dir2/file2
vvdn108@vvdn108:~/trainee/testdir$ svn delete http://192.168.10.10/svn/trainee/test/ntp-4.2.6p5/
      -m "Removing the unwanted directory"

Committed revision 1422.
vvdn108@vvdn108:~/trainee/testdir$
```

From WC

From URL

```
vvdn108@vvdn108: ~/trainee/testdir
vvdn108@vvdn108:~/trainee/testdir$ svn delete --keep-local dir1/file1
D      dir1/file1
vvdn108@vvdn108:~/trainee/testdir$
```

Deletes only from Repo

Other Useful SVN Commands

- **svn commit (ci)** — Send changes from your working copy to the repository.
- Until we commit changes i.e. adding a file , deleting etc they won't be reflected on Version control system.

```
vvdn108@vvdn108: ~/trainee/testdir
vvdn108@vvdn108:~/trainee/testdir$ svn commit -m "Committing changes"
Deleting      dir2

Committed revision 1423.
vvdn108@vvdn108:~/trainee/testdir$
```

Other Useful SVN Commands

- **svn update (up)** — Update your working copy.

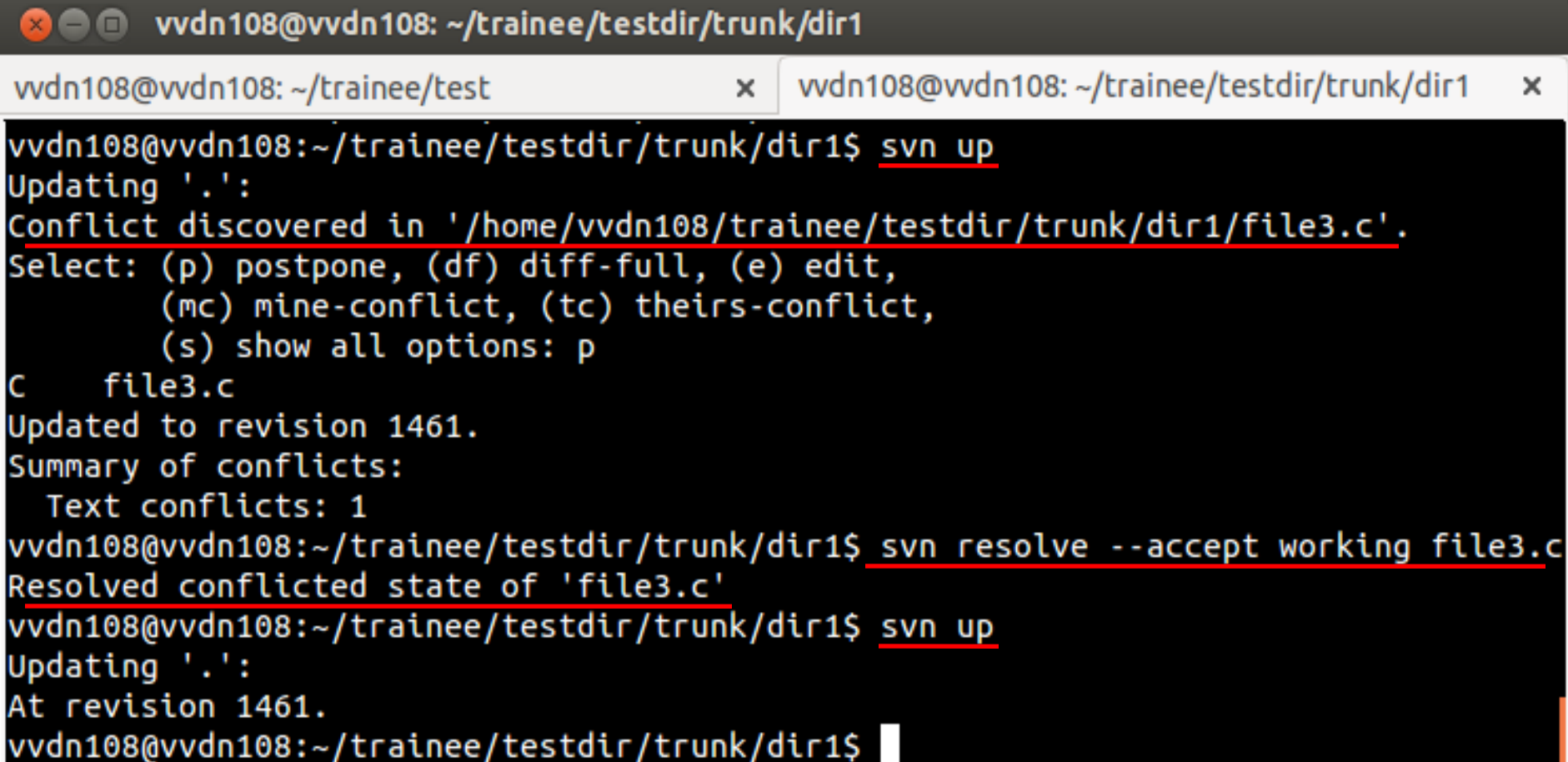
```
vvdn108@vvdn108: ~/trainee/test
vvdn108@vvdn108:~/trainee/test$ svn update
Updating '.':
D    ntp-4.2.6p5
Updated to revision 1423.
vvdn108@vvdn108:~/trainee/test$
```

- Otherwise, it synchronizes the working copy to the revision given by the `--revision (-r)` option.

```
vvdn108@vvdn108: ~/trainee/test
vvdn108@vvdn108:~/trainee/test$ svn update -r 1400
Updating '.':
D    .
Updated to revision 1400.
vvdn108@vvdn108:~/trainee/test$
```

Other Useful SVN Commands

- **svn resolve** — Resolve conflicts on working copy files or directories.



A terminal window titled 'vvdn108@vvdn108: ~/trainee/testdir/trunk/dir1' showing the execution of SVN commands. The user runs 'svn up', which reports a conflict in 'file3.c'. They select option 'p' (postpone) and then run 'svn resolve --accept working file3.c'. Finally, they run 'svn up' again, which completes successfully at revision 1461.

```
vvdn108@vvdn108:~/trainee/testdir/trunk/dir1$ svn up
Updating '.':
Conflict discovered in '/home/vvdn108/trainee/testdir/trunk/dir1/file3.c'.
Select: (p) postpone, (df) diff-full, (e) edit,
        (mc) mine-conflict, (tc) theirs-conflict,
        (s) show all options: p
C    file3.c
Updated to revision 1461.
Summary of conflicts:
  Text conflicts: 1
vvdn108@vvdn108:~/trainee/testdir/trunk/dir1$ svn resolve --accept working file3.c
Resolved conflicted state of 'file3.c'
vvdn108@vvdn108:~/trainee/testdir/trunk/dir1$ svn up
Updating '.':
At revision 1461.
vvdn108@vvdn108:~/trainee/testdir/trunk/dir1$
```


Other Useful SVN Commands

- **svn log** — Display commit log messages.

```
vvdn108@vvdn108: ~/trainee/test
vvdn108@vvdn108:~/trainee/test$ svn log -r 1422:1423
-----
r1422 | rakeshp | 2013-08-08 14:38:32 +0530 (Thu, 08 Aug 2013) | 1 line
Removing the unwanted directory
-----
vvdn108@vvdn108:~/trainee/test$
```

```
vvdn108@vvdn108: ~/trainee/test
vvdn108@vvdn108:~/trainee/test$ svn log http://192.168.10.10/svn/trainee/testdir
-----
r1423 | rakeshp | 2013-08-08 14:39:44 +0530 (Thu, 08 Aug 2013) | 1 line
Committing changes
-----
r1421 | rakeshp | 2013-08-08 14:36:06 +0530 (Thu, 08 Aug 2013) | 1 line
Added dir
-----
r1419 | rakeshp | 2013-08-08 14:09:55 +0530 (Thu, 08 Aug 2013) | 1 line
Added files
-----
```

Managing Project with SVN

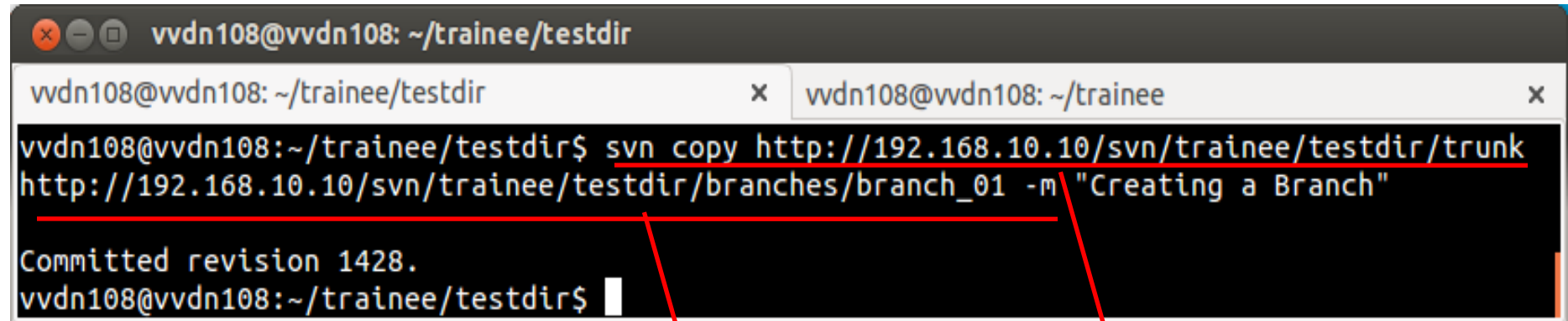
Trunks :

- Main place where stable code can be found.
- **NEVER** work directly on the trunk.
- **Unless** you have to deal with a bug which is quick and easy to fix.
- **Do not** make too many exceptions to the previous statement.
- ***Every other situation must imply the creation of a branch.***
- Do not commit changes (from a branch merge) to the trunk which may break it .

Managing Project with SVN

How to create your own Branch ?

- Your own branch or a line of development, in the project repository.



A terminal window titled 'vvdn108@vvdn108: ~/trainee/testdir' showing the execution of an SVN command. The command is 'svn copy http://192.168.10.10/svn/trainee/testdir/trunk http://192.168.10.10/svn/trainee/testdir/branches/branch_01 -m "Creating a Branch"'. The output is 'Committed revision 1428.' followed by a new prompt. Two red lines originate from the command: one points from the source URL to the 'Source Address' label, and another points from the destination URL to the 'Destination Address' label.

```
vvdn108@vvdn108: ~/trainee/testdir
vvdn108@vvdn108: ~/trainee/testdir$ svn copy http://192.168.10.10/svn/trainee/testdir/trunk
http://192.168.10.10/svn/trainee/testdir/branches/branch_01 -m "Creating a Branch"
Committed revision 1428.
vvdn108@vvdn108:~/trainee/testdir$
```

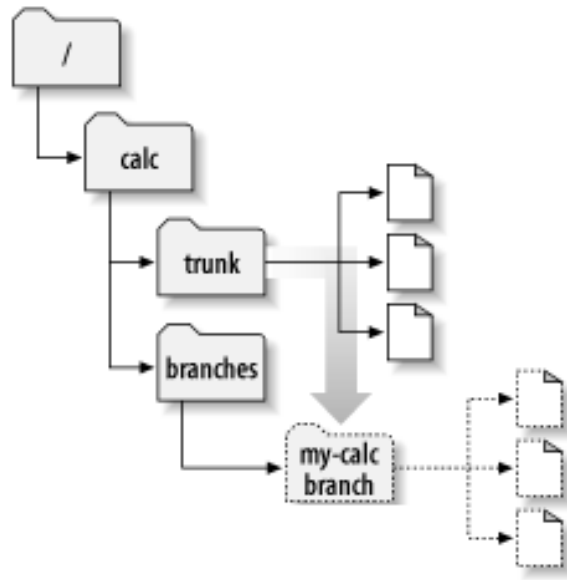
Destination
Address

Source Address

Managing Project with SVN

Cheap Copies:

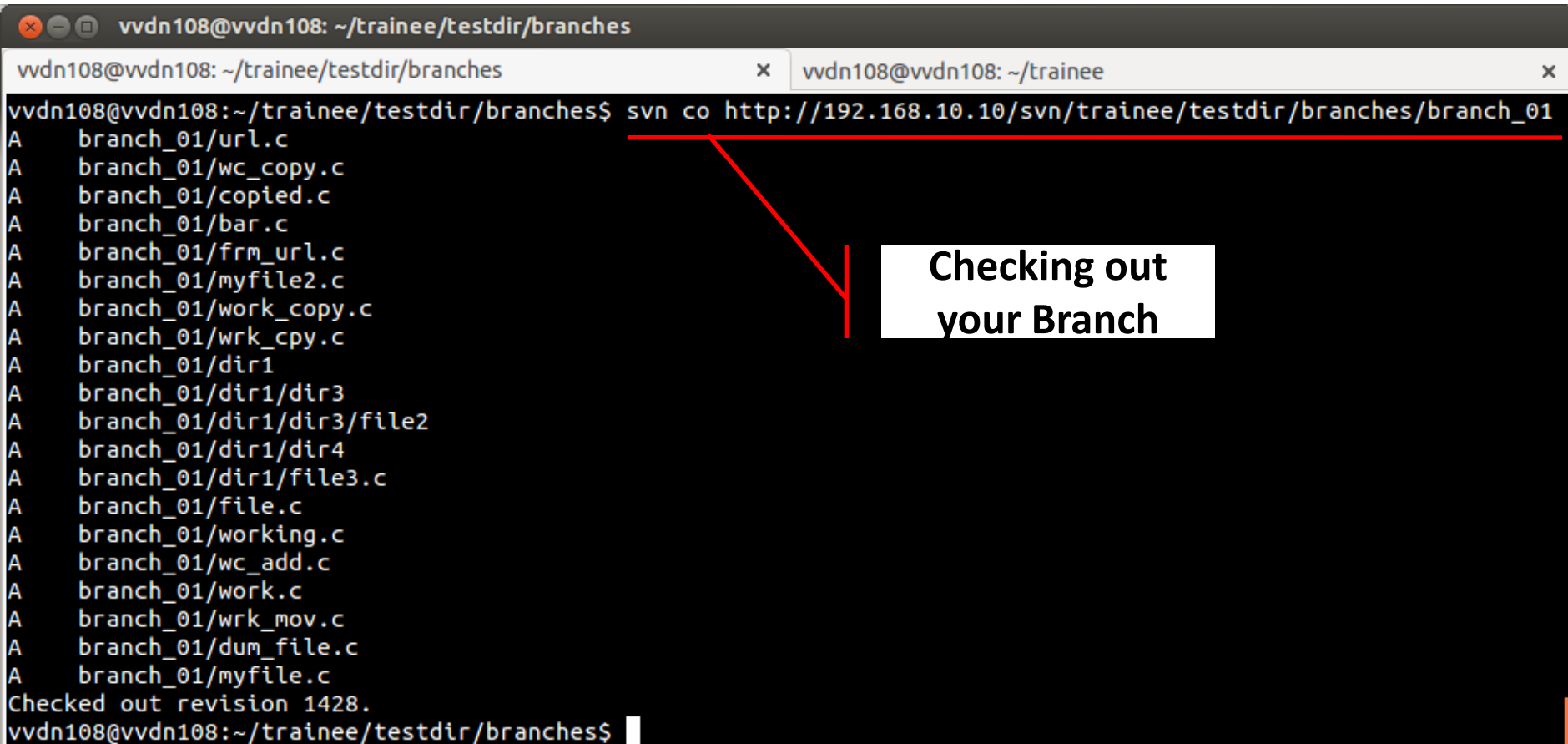
- Subversion doesn't actually duplicate any data.
- It creates a new directory entry that points to an *existing* tree (kind of hardlink).
- It duplicates data only when it is necessary to disambiguate different versions of objects.



Managing Project with SVN

- **Working with Your Branch**

- Check out a new working copy your branch
- Start using it .



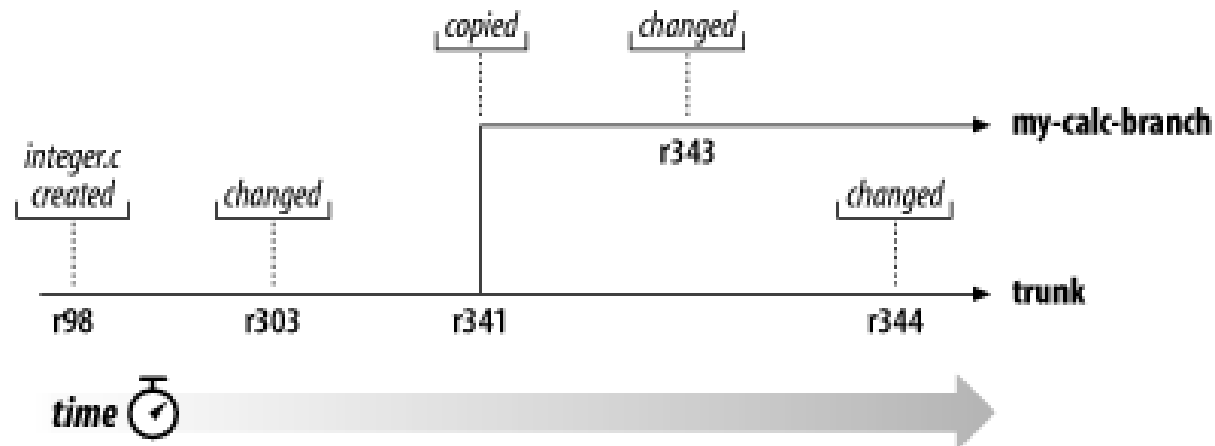
A terminal window with a dark background and light text. The title bar shows 'vvdn108@vvdn108: ~/trainee/testdir/branches'. The command prompt shows 'vvdn108@vvdn108: ~/trainee/testdir/branches\$'. The command entered is 'svn co http://192.168.10.10/svn/trainee/testdir/branches/branch_01'. The output lists 20 files and directories, each preceded by 'A'. The list ends with 'Checked out revision 1428.' and the prompt 'vvdn108@vvdn108:~/trainee/testdir/branches\$'. A red line underlines the URL in the command, and a red arrow points from this line to a white box containing the text 'Checking out your Branch'.

```
vvdn108@vvdn108: ~/trainee/testdir/branches
vvdn108@vvdn108: ~/trainee/testdir/branches$ svn co http://192.168.10.10/svn/trainee/testdir/branches/branch_01
A    branch_01/url.c
A    branch_01/wc_copy.c
A    branch_01/copied.c
A    branch_01/bar.c
A    branch_01/frm_url.c
A    branch_01/myfile2.c
A    branch_01/work_copy.c
A    branch_01/wrk_cpy.c
A    branch_01/dir1
A    branch_01/dir1/dir3
A    branch_01/dir1/dir3/file2
A    branch_01/dir1/dir4
A    branch_01/dir1/file3.c
A    branch_01/file.c
A    branch_01/working.c
A    branch_01/wc_add.c
A    branch_01/work.c
A    branch_01/wrk_mov.c
A    branch_01/dum_file.c
A    branch_01/myfile.c
Checked out revision 1428.
vvdn108@vvdn108:~/trainee/testdir/branches$
```

Managing Project using SVN

When things get interesting!!

- Changes made by you in your branch will not be visible to someone working on trunk.
- Similarly changes made on trunk won't be visible to you in branch.



Managing Project Using SVN

Basic Merging:

- Keeping your Branch in synch:

```
vvdn108@vvdn108: ~/trainee/testdir/branches/branch_01
vvdn108@vvdn108: ~/trainee/testdir/branches/branch_01 x vvdn108@vvdn108: ~/trainee x
vvdn108@vvdn108:~/trainee/testdir/branches/branch_01$ svn merge http://192.168.10.10/svn/trainee/testdir/trunk
--- Merging r1425 through r1432 into '.':
U    dum_file.c
```

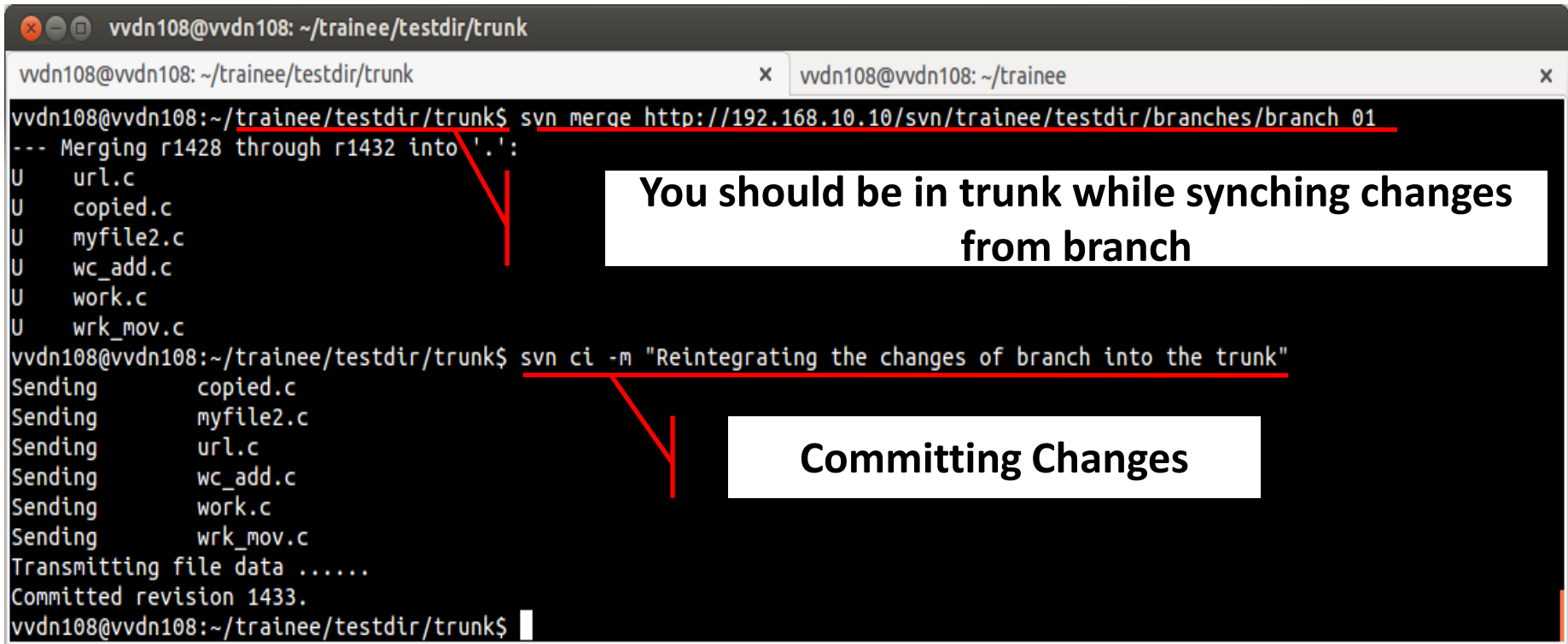
You should be in branch while synching changes from trunk

- ***Committing is must after merge for changes to reflect on repo .***

```
vvdn108@vvdn108:~/trainee/testdir/branches/branch_01$ svn ci -m "Reflecting changes done in trunk to the branch"
Sending      dum_file.c
Transmitting file data .
Committed revision 1436.
```

Managing Project Using SVN

- **Reintegrating a Branch:**
 - Bring your branch into sync with the trunk again.



The image shows a terminal window with two tabs. The active tab is titled 'vvdn108@vvdn108: ~/traineer/testdir/trunk'. The terminal output shows the execution of 'svn merge' and 'svn ci' commands. A red arrow points from the first callout box to the 'svn merge' command. Another red arrow points from the second callout box to the 'svn ci' command.

```
vvdn108@vvdn108: ~/traineer/testdir/trunk
vvdn108@vvdn108: ~/traineer/testdir/trunk$ svn merge http://192.168.10.10/svn/traineer/testdir/branches/branch_01
--- Merging r1428 through r1432 into '.':
U    url.c
U    copied.c
U    myfile2.c
U    wc_add.c
U    work.c
U    wrk_mov.c
vvdn108@vvdn108:~/traineer/testdir/trunk$ svn ci -m "Reintegrating the changes of branch into the trunk"
Sending        copied.c
Sending        myfile2.c
Sending        url.c
Sending        wc_add.c
Sending        work.c
Sending        wrk_mov.c
Transmitting file data .....
Committed revision 1433.
vvdn108@vvdn108:~/traineer/testdir/trunk$
```

You should be in trunk while syncing changes from branch

Committing Changes

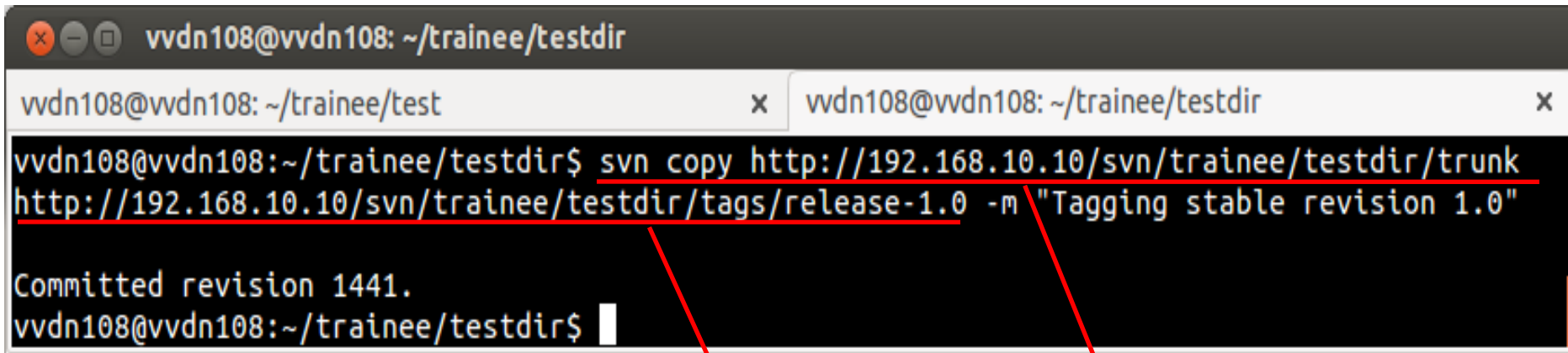
Once you have reintegrated your branch you may delete it.

Managing Project using SVN

Tag:

A tag is just a “snapshot” of a project in time.

- **Creating a Simple Tag**



```
vvdn108@vvdn108: ~/trainee/testdir
vvdn108@vvdn108: ~/trainee/test
vvdn108@vvdn108:~/trainee/testdir$ svn copy http://192.168.10.10/svn/trainee/testdir/trunk
http://192.168.10.10/svn/trainee/testdir/tags/release-1.0 -m "Tagging stable revision 1.0"
Committed revision 1441.
vvdn108@vvdn108:~/trainee/testdir$
```

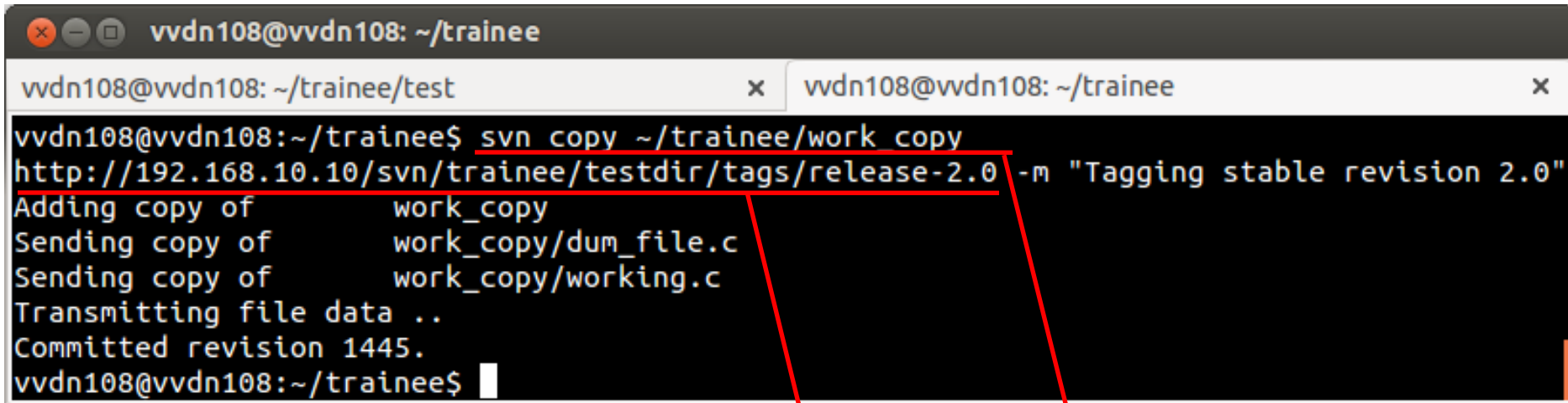
Source Addr:
Stable Trunk State
(Repo)

Destination Addr with
appropriate name

Managing Project using SVN

Tag:

- Creating a Complex Tag



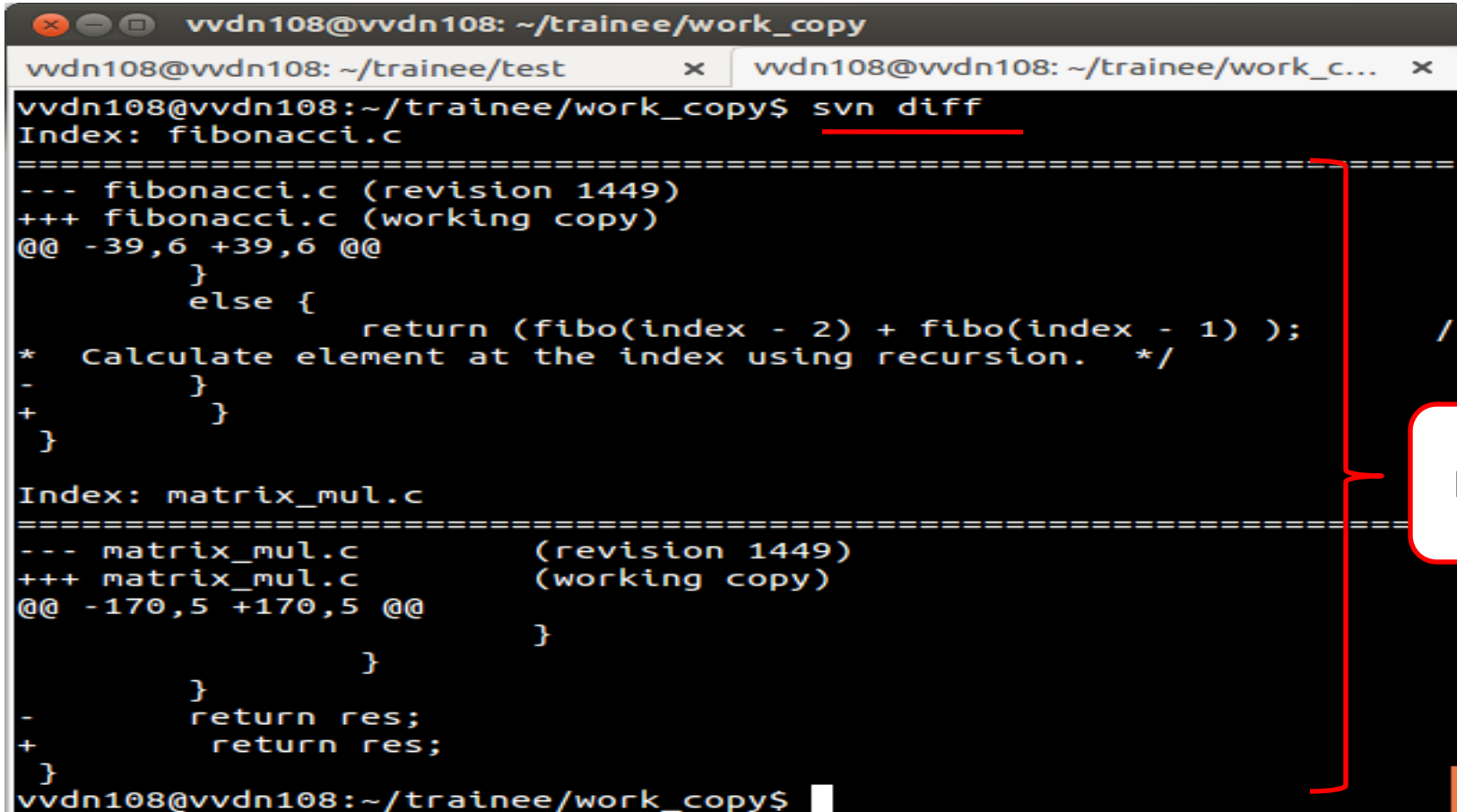
```
vvdn108@vvdn108: ~/trainee
vvdn108@vvdn108: ~/trainee/test x vvdn108@vvdn108: ~/trainee x
vvdn108@vvdn108:~/trainee$ svn copy ~/trainee/work copy
http://192.168.10.10/svn/trainee/testdir/tags/release-2.0 -m "Tagging stable revision 2.0"
Adding copy of          work_copy
Sending copy of          work_copy/dum_file.c
Sending copy of          work_copy/working.c
Transmitting file data ..
Committed revision 1445.
vvdn108@vvdn108:~/trainee$
```

Source Addr:
Stable Working
Copy State

Destination Addr with
appropriate name

Useful commands for Project management

- **svn diff (di)** — This displays the differences between two revisions or paths.
 - Examining local changes



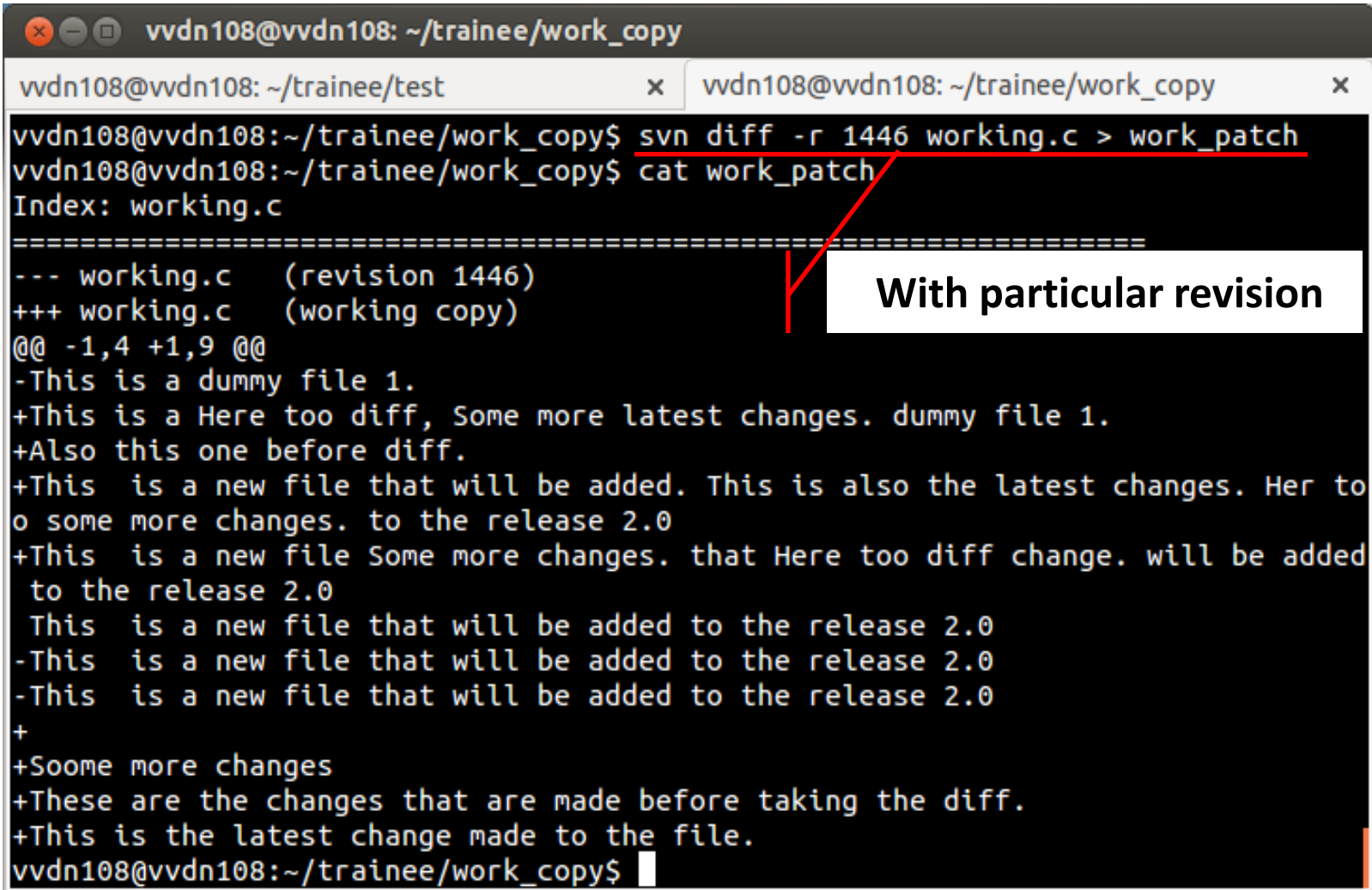
```
vvdn108@vvdn108: ~/traineer/work_copy
vvdn108@vvdn108: ~/traineer/test x vvdn108@vvdn108: ~/traineer/work_c... x
vvdn108@vvdn108:~/traineer/work_copy$ svn diff
Index: fibonacci.c
=====
--- fibonacci.c (revision 1449)
+++ fibonacci.c (working copy)
@@ -39,6 +39,6 @@
     }
     else {
         return (fibo(index - 2) + fibo(index - 1) );
     /* Calculate element at the index using recursion. */
     }
 }

Index: matrix_mul.c
=====
--- matrix_mul.c (revision 1449)
+++ matrix_mul.c (working copy)
@@ -170,5 +170,5 @@
     }
 }
 return res;
+ return res;
 }
vvdn108@vvdn108:~/traineer/work_copy$
```

Uni
Format
Diff

Useful commands for Project management

- Comparing your working copy to the repository



```
vvdn108@vvdn108: ~/trainee/work_copy
vvdn108@vvdn108: ~/trainee/test x vvdn108@vvdn108: ~/trainee/work_copy x
vvdn108@vvdn108:~/trainee/work_copy$ svn diff -r 1446 working.c > work_patch
vvdn108@vvdn108:~/trainee/work_copy$ cat work_patch
Index: working.c
=====
--- working.c      (revision 1446)
+++ working.c      (working copy)
@@ -1,4 +1,9 @@
-This is a dummy file 1.
+This is a Here too diff, Some more latest changes. dummy file 1.
+Also this one before diff.
+This is a new file that will be added. This is also the latest changes. Her to
o some more changes. to the release 2.0
+This is a new file Some more changes. that Here too diff change. will be added
to the release 2.0
+This is a new file that will be added to the release 2.0
-This is a new file that will be added to the release 2.0
-This is a new file that will be added to the release 2.0
+
+Soome more changes
+These are the changes that are made before taking the diff.
+This is the latest change made to the file.
vvdn108@vvdn108:~/trainee/work_copy$
```

With particular revision

Useful commands for Project management

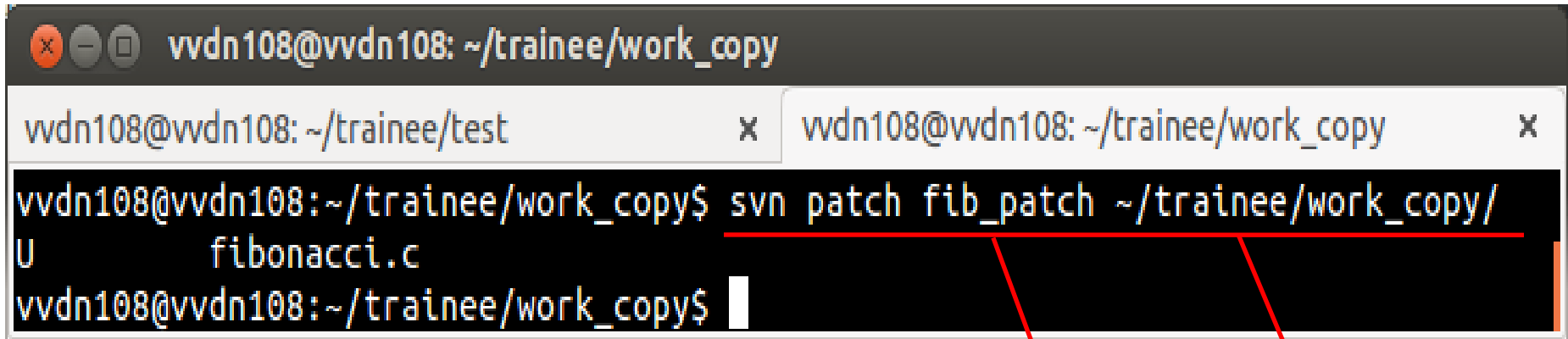
- Comparing repository revisions.

```
vvdn108@vvdn108: ~/traineer/work_copy
vvdn108@vvdn108: ~/traineer/test x vvdn108@vvdn108: ~/traineer/work_copy x
vvdn108@vvdn108:~/traineer/work_copy$ svn diff -r 1446:1448 working.c > rev_patch
vvdn108@vvdn108:~/traineer/work_copy$ cat rev_patch
Index: working.c
=====
--- working.c      (revision 1446)
+++ working.c      (revision 1448)
@@ -1,4 +1,8 @@
-This is a dummy file 1.
+This is a Here too diff, dummy file 1.
+Also this one before diff.
+This is a new file that will be addedHer too some more changes. to the release 2
.0
+This is a new file Some more changes. that Here too diff change. will be added t
o the release 2.0
+This is a new file that will be added to the release 2.0
-This is a new file that will be added to the release 2.0
-This is a new file that will be added to the release 2.0
+
+Soome more changes
+These are the changes that are made before taking the diff.
vvdn108@vvdn108:~/traineer/work_copy$
```

Comparing revisions and
redirecting output

Useful commands for Project management

- **svn patch** — Apply changes represented in a unidiff patch to the working copy.



A terminal window with a dark background and light-colored text. The title bar shows the user 'vvdn108' and the current directory '~/traineer/work_copy'. The terminal has two tabs: 'vvdn108@vvdn108: ~/traineer/test' and 'vvdn108@vvdn108: ~/traineer/work_copy'. The active tab shows the command 'svn patch fib_patch ~/traineer/work_copy/' being entered. Below the command, the output 'U fibonacci.c' is visible. A red horizontal line underlines the command, and two red arrows point from the labels 'Patchfile' and 'WC path' to 'fib_patch' and '~/traineer/work_copy/' respectively.

```
vvdn108@vvdn108: ~/traineer/work_copy
vvdn108@vvdn108: ~/traineer/test
vvdn108@vvdn108: ~/traineer/work_copy$ svn patch fib_patch ~/traineer/work_copy/
U      fibonacci.c
vvdn108@vvdn108: ~/traineer/work_copy$
```

Patchfile

WC path

Reversing the changes made by Patch

```
vvdn108@vvdn108: ~/trainee/test
vvdn108@vvdn108:~/trainee/test$ cat patch_demo.c
#include<stdio.h>
main()
{
    getchar();
}
vvdn108@vvdn108:~/trainee/test$ cat demo.patch
Index: patch_demo.c
=====
--- patch_demo.c      (revision 1479)
+++ patch_demo.c      (working copy)
@@ -1,5 +1,5 @@
-#include<stdio.h>
+#include<string.h>
main()
{
-    getchar();
+    printf("Changes made.\n");
}
vvdn108@vvdn108:~/trainee/test$ svn patch demo.patch ~/trainee/test/
U      patch_demo.c
vvdn108@vvdn108:~/trainee/test$ cat patch_demo.c
#include<string.h>
main()
{
    printf("Changes made.\n");
}
vvdn108@vvdn108:~/trainee/test$ svn patch --reverse-diff demo.patch ~/trainee/test/
G      patch_demo.c
vvdn108@vvdn108:~/trainee/test$ cat patch_demo.c
#include<stdio.h>
main()
{
    getchar();
}
vvdn108@vvdn108:~/trainee/test$
```

Before patch

PATCH FILE

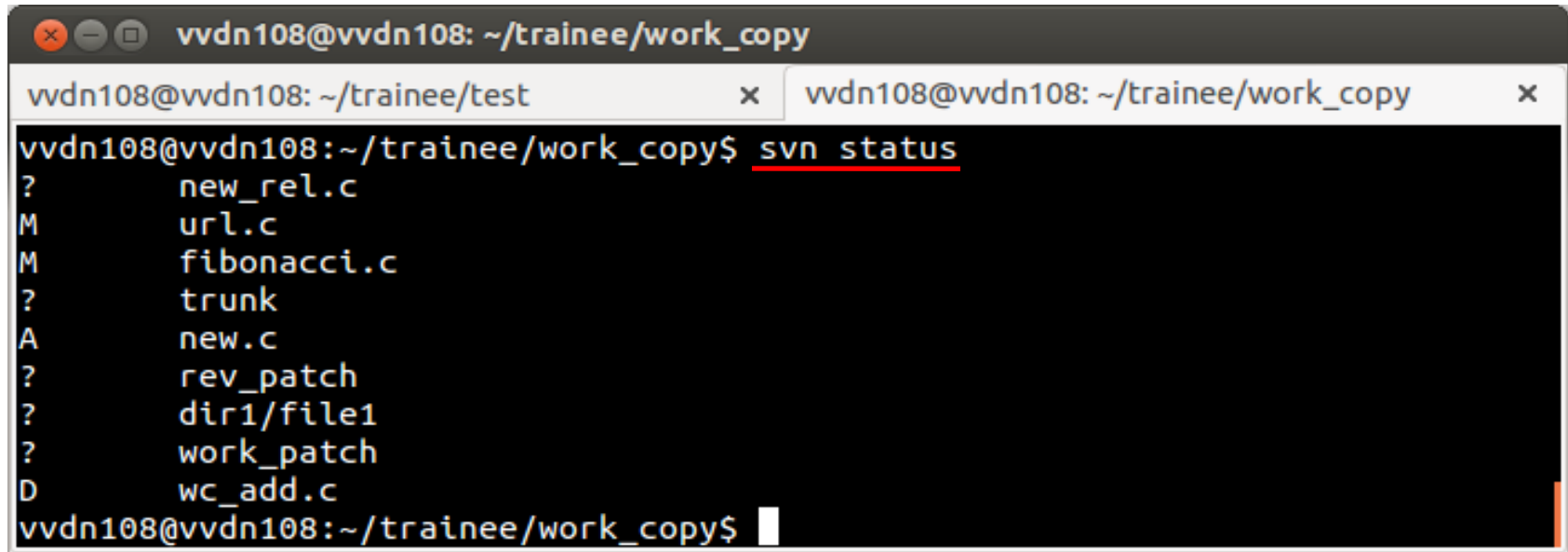
After patch

With --reverse-diff

After reversing patch

Useful commands for Project management

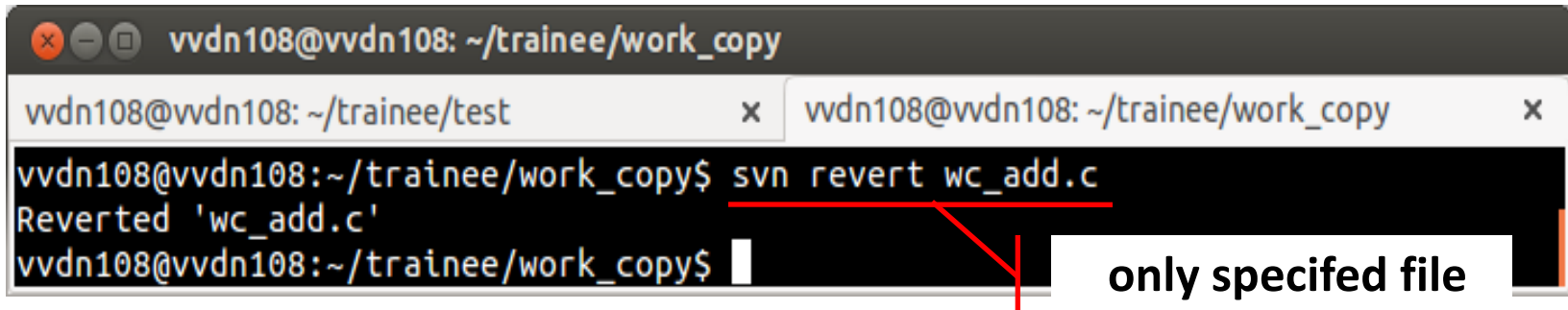
- **svn status (stat, st)** — Print the status of working copy files and directories.



```
vvdn108@vvdn108: ~/trainee/work_copy
vvdn108@vvdn108: ~/trainee/test x vvdn108@vvdn108: ~/trainee/work_copy x
vvdn108@vvdn108:~/trainee/work_copy$ svn status
?      new_rel.c
M      url.c
M      fibonacci.c
?      trunk
A      new.c
?      rev_patch
?      dir1/file1
?      work_patch
D      wc_add.c
vvdn108@vvdn108:~/trainee/work_copy$
```


Useful commands for Project management

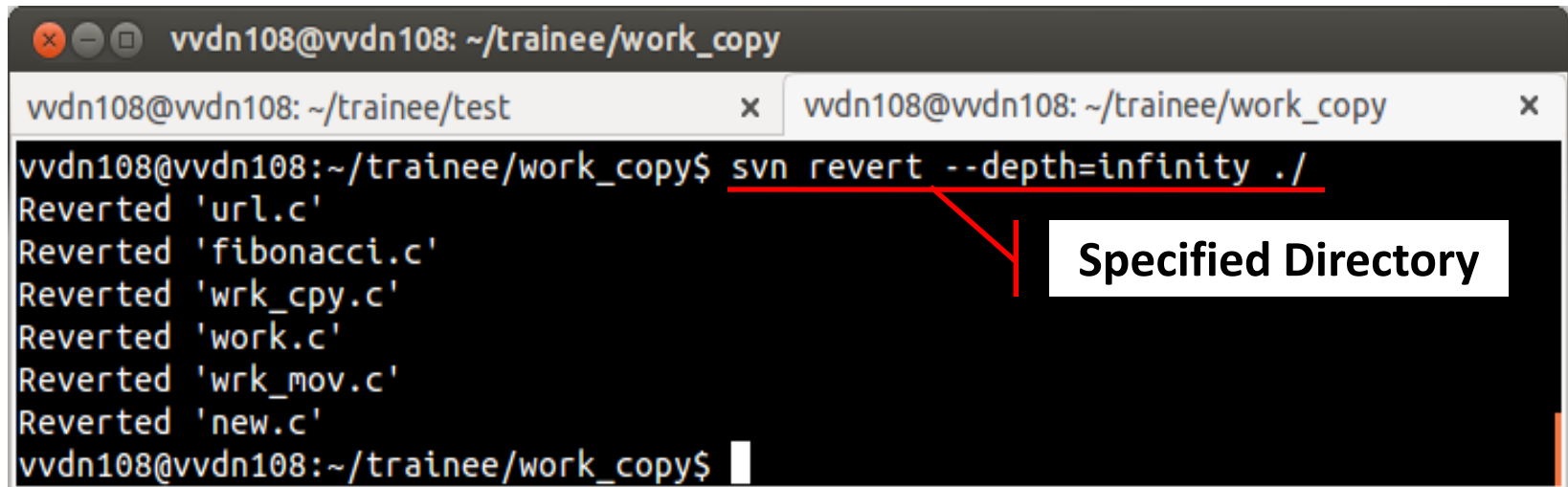
- **svn revert** — Undo all local edits.



A terminal window titled 'vvdn108@vvdn108: ~/trainee/work_copy' with two tabs: 'vvdn108@vvdn108: ~/trainee/test' and 'vvdn108@vvdn108: ~/trainee/work_copy'. The command 'svn revert wc_add.c' is entered and executed. The output is 'Reverted 'wc_add.c''. A red line points from the text 'only specified file' to the file name 'wc_add.c' in the command.

```
vvdn108@vvdn108: ~/trainee/work_copy
vvdn108@vvdn108: ~/trainee/test x vvdn108@vvdn108: ~/trainee/work_copy x
vvdn108@vvdn108:~/trainee/work_copy$ svn revert wc_add.c
Reverted 'wc_add.c'
vvdn108@vvdn108:~/trainee/work_copy$
```

only specified file



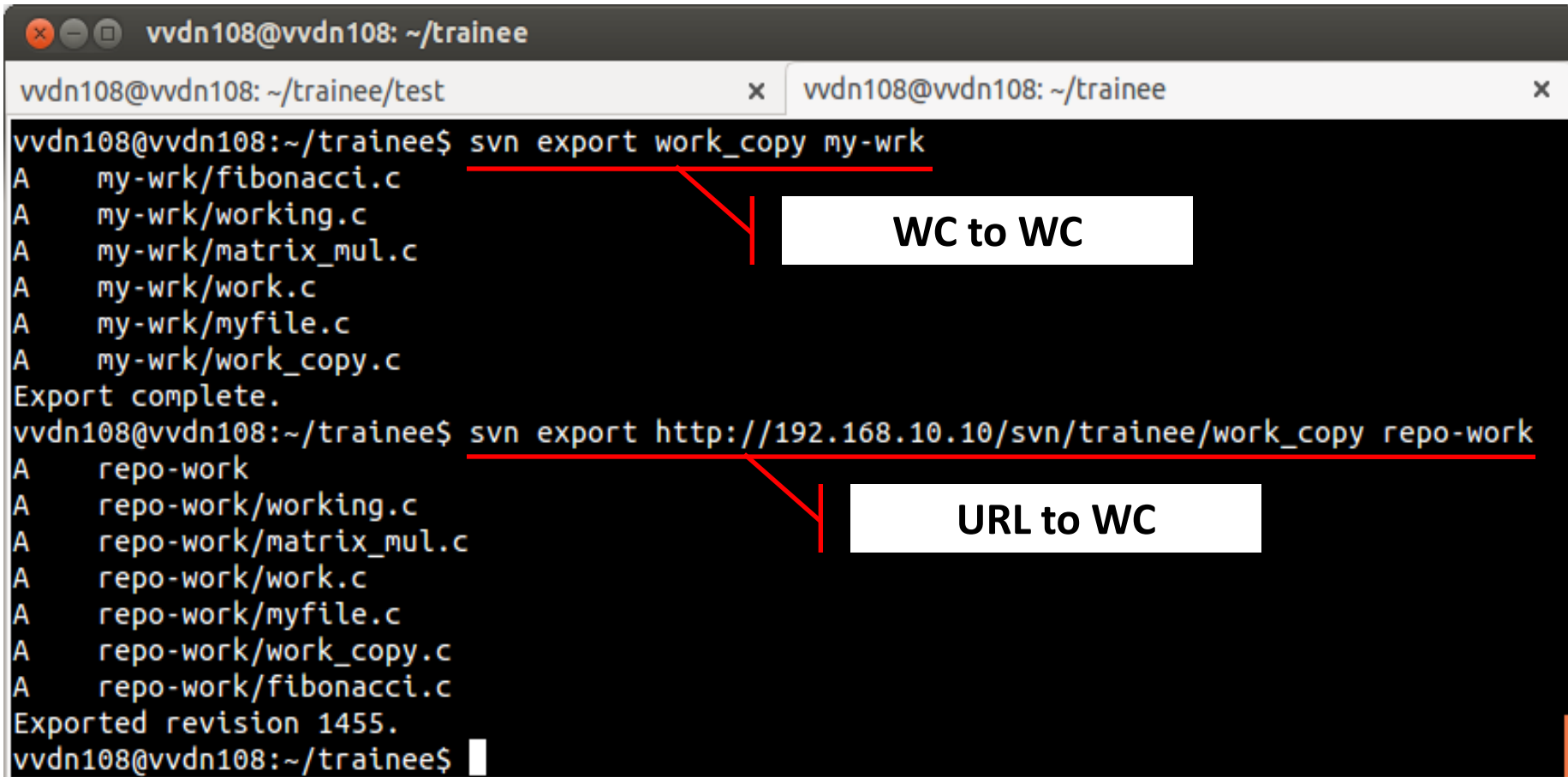
A terminal window titled 'vvdn108@vvdn108: ~/trainee/work_copy' with two tabs: 'vvdn108@vvdn108: ~/trainee/test' and 'vvdn108@vvdn108: ~/trainee/work_copy'. The command 'svn revert --depth=infinity .' is entered and executed. The output lists several files that have been reverted: 'url.c', 'fibonacci.c', 'wrk_cpy.c', 'work.c', 'wrk_mov.c', and 'new.c'. A red line points from the text 'Specified Directory' to the '.' in the command.

```
vvdn108@vvdn108: ~/trainee/work_copy
vvdn108@vvdn108: ~/trainee/test x vvdn108@vvdn108: ~/trainee/work_copy x
vvdn108@vvdn108:~/trainee/work_copy$ svn revert --depth=infinity .
Reverted 'url.c'
Reverted 'fibonacci.c'
Reverted 'wrk_cpy.c'
Reverted 'work.c'
Reverted 'wrk_mov.c'
Reverted 'new.c'
vvdn108@vvdn108:~/trainee/work_copy$
```

Specified Directory

Useful commands for Project management

- **svn export** — Export a clean directory tree.



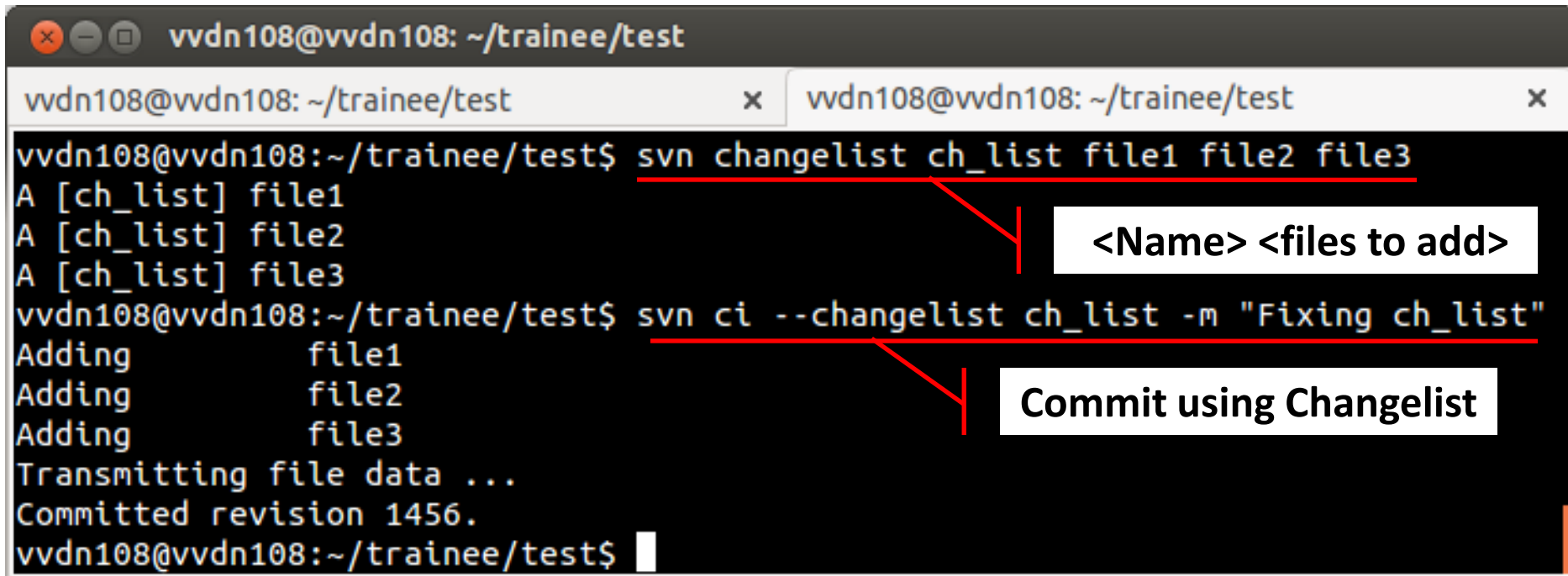
The image shows a terminal window with two tabs. The first tab is titled 'vvdn108@vvdn108: ~/traineetest' and the second is 'vvdn108@vvdn108: ~/traineet'. The terminal content is as follows:

```
vvdn108@vvdn108:~/traineet$ svn export work_copy my-wrk
A    my-wrk/fibonacci.c
A    my-wrk/working.c
A    my-wrk/matrix_mul.c
A    my-wrk/work.c
A    my-wrk/myfile.c
A    my-wrk/work_copy.c
Export complete.
vvdn108@vvdn108:~/traineet$ svn export http://192.168.10.10/svn/traineet/work_copy repo-work
A    repo-work
A    repo-work/working.c
A    repo-work/matrix_mul.c
A    repo-work/work.c
A    repo-work/myfile.c
A    repo-work/work_copy.c
A    repo-work/fibonacci.c
Exported revision 1455.
vvdn108@vvdn108:~/traineet$
```

Two red arrows point from text boxes to the commands. The first arrow points from a box labeled 'WC to WC' to the command 'svn export work_copy my-wrk'. The second arrow points from a box labeled 'URL to WC' to the command 'svn export http://192.168.10.10/svn/traineet/work_copy repo-work'.

Useful commands for Project management

- **svn changelist (cl)** — Associate (or deassociate) local paths with a changelist.



The image shows a terminal window with two tabs, both titled 'vvdn108@vvdn108: ~/trainee/test'. The terminal output is as follows:

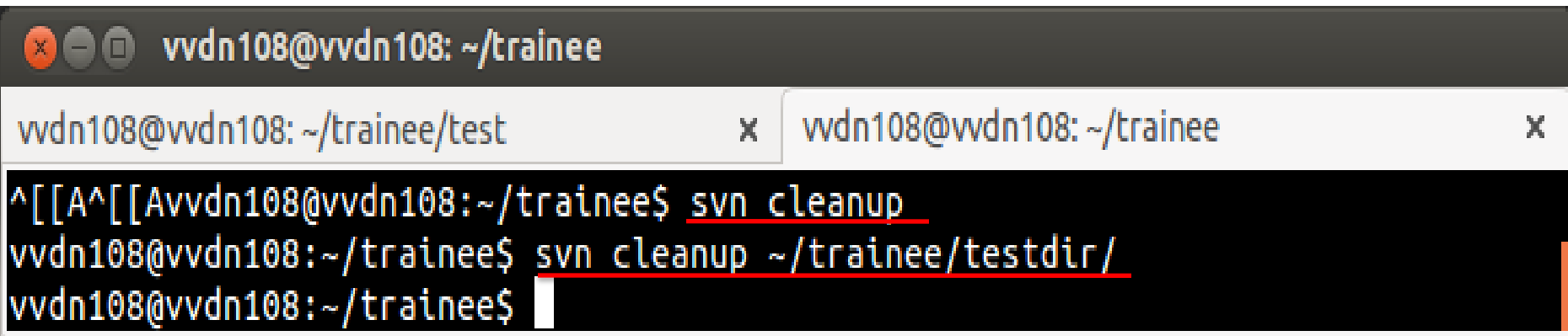
```
vvdn108@vvdn108:~/trainee/test$ svn changelist ch_list file1 file2 file3
A [ch_list] file1
A [ch_list] file2
A [ch_list] file3
vvdn108@vvdn108:~/trainee/test$ svn ci --changelist ch_list -m "Fixing ch_list"
Adding          file1
Adding          file2
Adding          file3
Transmitting file data ...
Committed revision 1456.
vvdn108@vvdn108:~/trainee/test$
```

Two red arrows point from white callout boxes to the underlined commands:

- The first arrow points from the box containing **<Name> <files to add>** to the command `svn changelist ch_list file1 file2 file3`.
- The second arrow points from the box containing **Commit using Changelist** to the command `svn ci --changelist ch_list -m "Fixing ch_list"`.

Useful commands for Project management

- **svn cleanup** — Recursively clean up the working copy, removing working copy locks and resuming unfinished operations.



A terminal window with a dark background and light text. The title bar shows 'vvdn108@vvdn108: ~/trainee'. There are two tabs: 'vvdn108@vvdn108: ~/trainee/test' and 'vvdn108@vvdn108: ~/trainee'. The main content area shows the following commands and output:

```
vvdn108@vvdn108:~/trainee$ svn cleanup
vvdn108@vvdn108:~/trainee$ svn cleanup ~/trainee/testdir/
vvdn108@vvdn108:~/trainee$
```

Some Useful Practices while using SVN

- **Follow a consistent directory structure.**
- The Subversion project officially recommends :
 - a project root representing an anchoring point for project.
 - three subdirectories: /trunk, /branches, and /tags.
- A repository may contain only one project root, or no. of them.

```
Project1/  
  branches/  
  tags/  
  trunk/  
Project2/  
  branches/  
  tags/  
  trunk/
```

Some Useful Practices while using SVN

General practices:

- **Single commit should only reflect a single purpose**
 - Fixing of a specific bug
 - Addition of a new feature
 - or some particular task.
- **Try not to resolve multiple issues with single commit.**
- **Detailed Commit messages are helpful to others.**

Some Useful Practices while using SVN

General practices:

- **Certain commits are disallowed**
 - Cannot commit deletion of a file or directory which doesn't have working revision HEAD.
 - Cannot commit property change to directory which doesn't have working revision HEAD.
- **Should not commit binaries (executables) to SVN repo.**
- **svn update before commit changes to avoid conflicts .**
- **Commit often & Update very often.**

Some Useful Practices while using SVN

General practices:

- Whenever you need to **copy, delete, move or rename** files or folders in repository.
- Do so using **corresponding file operations** in the version control system.
- If done only on local file system **the history of changes will be lost forever.**

Some Useful Practices while using SVN

Branching Best Practices

- **Should be created for new features & major bugs.**
- **Commit often**
 - Commit small changes frequently
 - Rather than committing many small changes at once in chunk.
 - Reduces merge conflicts
- **Delete unwanted branches –**
 - A successful merge has been performed.
 - A successful reintegration merge has been performed

Some Useful Practices while using SVN

Merging Best Practices:

- **Start with a checkout or an update.**
- **Merge on logical checkpoints**
 - On reaching a certain level of stability and maturity.
 - Never merge when an experimental change has been made.
- **Use log messages**
- **Merge soon**
 - So that the team remains updated with changes you made.
- **Run Frequent 'SVN Updates'**
 - So that you stay updated with changes made in the trunk.

Some Useful Practices while using SVN

Tagging Best Practices:

- Before merges
- Tag major releases
- Before making sweeping changes
- Use proper tag names

References

- <http://svnbook.red-bean.com/nightly/en/index.html>
- [http://en.wikipedia.org/wiki/Apache Subversion](http://en.wikipedia.org/wiki/Apache_Subversion)
- <http://codefez.com/creating-projects-in-subversion-trunk-tags-branches/>
- <http://blog.jmfeurprier.com/2010/02/08/svn-trunk-branches-and-tags/>
- <http://jaxenter.com/tutorial-apache-subversion-best-practices-46037.html>