



Source control with Subversion A user perspective

Aaron Ponti

What is Subversion?

- It is a free and open-source version control system
- It manages files and directories, and the changes made to them, over time:
 - It's a "time machine"
 - It allows you to recover older versions of your data or examine the history of how your data changed
 - It works with any type of files but gives its best with text files
 We will see later why...
- It fosters collaboration by allowing various people to modify and manage the same set of data from their respective locations
 - > Although it can be a perfect tool also for the lone user

Fundamental Subversion concepts

Repository

- The central data store
- Versioning
 - The full history of a repository
- Working copy
 - The user's local version of the repository content

Commits

The changes published to the repository

Revisions

The temporal snapshots of the repository

The Subversion repository

- At the core of Subversion is a repository, which is a central store of data
- The repository stores information in the form of a *filesystem tree*, a typical hierarchy of files and directory



- Any number of *clients* (users) connect to the repository, and then read or write to these files and folders:
 - By writing data, a client makes the information available to others
 - By reading data, the client receives information from others

Versioning

- What makes the Subversion repository special is that it remembers every change ever written to it (versioning):
 - every change to every file, and even changes to the directory tree itself, such as the addition, deletion, and rearrangement of files and directories
- When a client reads from the repository, it not only sees the latest version of the file system tree, but it also has the ability to view *all previous* states:
 - For example, a client can ask historical questions such as "What did this directory contain last Wednesday?" and "Who was the last person to change this file, and what changes did he make?"

Versioning models

All version control systems have to solve the same fundamental problem: how will the system allow users to share information, but prevent them from accidentally stepping on each other's feet?



The problem to avoid

The Lock-Modify-Unlock Solution

In this model, the repository allows only one person to change a file at a time: this exclusivity policy is managed using locks



Sally cannot modify the file until Harry has released the lock

The Copy-Modify-Merge Solution

- In this model^(*), each user's client contacts the project repository and creates a personal working copy (a local copy of the repository's files and directories)
- Users then work simultaneously and independently, modifying their private copies
- Finally, the private copies are merged together into a new, final version

The version control system often assists with the merging, but ultimately, a human being is responsible for making it happen correctly!

(*) This is the *main* model used by Subversion. Subversion also supports *locking* for those cases where merging or conflict-solving is impossible, e.g. when modifying binary files (e.g. artwork, sound files, ...)

The Copy-Modify-Merge Solution

- The copy-modify-merge model may sound a bit chaotic, but in practice, it runs extremely smoothly
- Users can work in parallel, never waiting for one another
- When they work on the same files, it turns out that most of their concurrent changes don't overlap at all: conflicts are infrequent
- The amount of time it takes to resolve conflicts is usually far less than the time lost by a locking system

Communication is the best way to avoid conflicts!

The working copy

- A Subversion working copy is an ordinary directory tree on your local system containing a collection of files and reflecting the content of the repository
- You can edit the files and directories in your working copy however you wish
- After you've made some changes to the files in your working copy, you commit your changes to the repository
 - the changes are now visible for the other people working with you on your project
- If other people commit their own changes, you will get them by updating your working copy

Commits and revisions

- A commit operation publishes changes to any number of files and directories as a single atomic transaction
 - i.e. Subversion guarantees that a commit sends all changes or none.
- Each time the repository accepts a commit, this creates a new state of the filesystem tree, called a *revision*
 - ▶ each revision is assigned a unique natural number, one greater than the number of the previous revision: unlike most version control systems, Subversion's revision numbers apply to entire trees, not individual files
 → global revision numbers



Subversion in action

Initial steps:

- Create a repository
- Import files
- Check out to a local working copy
- Basic work cycle:
 - Update your local copy
 - Modify your local copy
 - Add or delete files and directories
 - See (your) changes
 - Revert changes
 - Commit your local changes to the repository

Advanced actions:

- Solve conflicts
- Create and switch to tags/branches
- Merge branches

Subversion clients

Client	Platform	Modality	License	URL
Subversion	Windows, Mac OS X, Linux	Command line tool (and library)	Open source	http://subversion.apache.org/
TortoiseSVN	Windows	Graphical user interface	Open source	http://tortoisesvn.net/
eSVN	Windows, Mac OS X, Linux	Graphical user interface	Open source	http://zoneit.free.fr/esvn/index.php
RapidSVN	Windows, Mac OS X, Linux	Graphical user interface	Open source	http://www.rapidsvn.org/index.php/Main_Page
KDESvn	Linux (KDE)	Graphical user interface	Open source	http://kdesvn.alwins-world.de/
Subclipse	Eclipse	Graphical user interface	Open source	http://subclipse.tigris.org/
SmartSVN	Windows, Mac OS X, Linux	Graphical user interface	Commercial	http://www.syntevo.com/smartsvn/index.html
Versions	Mac OS X	Graphical user interface	Commercial	http://versionsapp.com/

Subversion command-line tool



🔤 C:\WINDOWS\system32\cmd.exe

E:\SubversionCourse>sun help usage: sun <subcommand> [options] [args] Subversion command-line client, version 1.6.12. Type 'sun help <subcommand>' for help on a specific subcommand. Type 'sun --version' to see the program version and RA modules or 'sun --version --quiet' to see just the version number.

Most subcommands take file and/or directory arguments, recursing on the directories. If no arguments are supplied to such a command, it recurses on the current directory (inclusive) by default.

Available subcommands:

```
add
blame (praise, annotate, ann)
cat
changelist (cl)
checkout (co)
cleanup
commit (ci)
copy (cp)
delete (del, remove, rm)
diff (di)
export
help (?, h)
import
info
list (ls)
lock
log
merge
mergeinfo
mkdir
move (mu, rename, ren)
propdel (pdel, pd)
propedit (pedit, pe)
propget (pget, pg)
proplist (plist, pl)
propset (pset, ps)
resolve
resolved
revert
status (stat, st)
switch (sw)
unlock
update (up)
```

Subversion is a tool for version control. For additional information, see http://subversion.tigris.org/ Subversion is a command-line tool and requires some typing in the Windows console! _ 🗆 ×

To get usage information type: svn help

or

svn help <command>
at the prompt

TortoiseSVN, a Windows shell extension



"The coolest Interface to (Sub)Version Control"

		Arrange Icons By 🔹 🕨 Refresh
		Paste Paste Shortcut
		NVIDIA Control Panel
	🛃 SVN Checkout 😤 Tortoise SVN 💦 🕨 🕨	
		New 🕨
s some		nView Properties
naows menu		Properties
1. 1.99\		

TortoiseSVN adds some entries to the Windows Explorer context menu (i.e. to the "right-click")

Subversion in action

- Initial steps:
 - Create a repository
 - Import files
 - Check out to a local working copy
- Basic work cycle:
 - Update your local copy
 - Modify your local copy
 - Add or delete files and directories
 - See (your) changes
 - Revert changes

Commit your local changes to the repository

Advanced actions:

- Solve conflicts
- Create and switch to tags/branches
- Merge branches

~95% of your time

Subversion in action

Initial steps:

- Create a repository \rightarrow most likely done by IT, we will skip if for now
- Import files \rightarrow optional step, we will skip it for now

Check out to a local working copy (and inspect it)

- Basic work cycle:
 - Update your local copy
 - Modify your local copy
 - Add or delete files and directories
 - See (your) changes
 - Revert changes
 - Commit your local changes to the repository
- Advanced actions:
 - Solve conflicts
 - Create tags; create/merge branches; switch to branches
 - Change properties

Subversion repository URLs

Schema	Access method
file:///	Direct repository access (on local disk)
http://	Access via WebDAV protocol to Subversion-aware Apache server
https://	Same as http://, but with SSL encryption.
svn://	Access via custom protocol to an synserve server
svn+ssh://	Same as svn://, but through an SSH tunnel.

file://

 can only be used to access a repository on the local disk; useful to keep a history of your own documents

http:// and https://

• the repository is on a remote Apache web server and can be accessed from many clients over the network: this is the most powerful configuration (and what we use at the FMI)

svn:// and svn+ssh://

the repository is on a remote server and can be accessed from many clients over the network; synserve is a simple server bundled with the subversion tools





	🐮 Checkout 🔀
We	Repository URL of repository: Image: http://source.fmi.ch/svn/FAIM/Courses/SourceControl_Subversion/trunk Checkout directory: E:\SubversionCourse\SourceControl_Subversion
checkout	Checkout Depth
trunk;	Fully recursive
more to this later	Omit e <u>x</u> ternals
	Revision
	HEAD revision
	C Revision Show log
Latest version of all files	OK Cancel Help

S U B V E R S I O N

svn checkout http://source.fmi.ch/svn/FAIM/Courses/SourceControl_Subversion/trunk E:\SubversionCourse\SourceControl_Subversion

Š

🕈 trunk - Ch	eckout - TortoiseS¥N Finished!	
Action	Path	Mime type
Command Added Added Added Added Added Added Completed	Checkout from http://source.fmi.ch/svn/FAIM/Courses/Source E:\SubversionCourse\SourceControl_Subversion\tests E:\SubversionCourse\SourceControl_Subversion\demo E:\SubversionCourse\SourceControl_Subversion\demo\accs.h E:\SubversionCourse\SourceControl_Subversion\demo\accs.c E:\SubversionCourse\SourceControl_Subversion\demo\accs.c E:\SubversionCourse\SourceControl_Subversion\demo\accs.c	eControl_Subversion/trunk, revision HEAD, Fully recursive, Externals included
L (atest repository revision HEAD) is 3	The repository contains following files and directories in the HEAD revision
, Added:6		<u>QK</u> Cancel

Our working copy now has the same content as the repository at its HEAD revision

- A checkout can occur *at any level* in the tree structure of the repository, and not necessarily at the root level:
 - In fact, it is very common **not** to check out the root of the repository: we will discuss this later
 - In analogy, updates and commits can also occur at all levels: for instance, one can decide to commit just one file



TortoiseSVN uses little overlay icons to represent the different possible states of our working copy



? Help

n 🕹 About

svn log

SUBVERSION



🗱 main.cpp Revision 3 - TortoiseMerge				
File Edit Navigate View Help				
<u>: ⊘</u>				
main.cpp Revision 2	^	mai	n.cpp Revision 3	×.
1/* 2 This is an example code to play with Subversion 3		1/* 2 This is an examp! 3	le code to play with	n Subversion
4 Aaron Ponti, 2001/01/05 5 */ 6		4 Source control with the second	ith Subversion cours 1/01/05	
<pre>7 #include "accs.h" 8 9 int main(int argc, char **argv) 10 { 11 // Display Hello World! 12 display(); 13 14 // Return success 15 return 0; 16 }</pre>		8 #include "accs.h" 9 10 int main(int argc, o 11 { 12 // Display Hello 13 display(); 14 15 // Return success 16 return 0; 17 }	char **argv) World!	This line was added
	×	I		
For Help, press F1. Scroll horizontally with Ctrl-Scrollwheel		Left View: ASCII CRLF Right View	w: ASCII CRLF / + 1	onflicts: 0 CAP NUM SCRL

TortoiseSVN provides a tool to visualize the changes in a file between any two revisions



About

<pre># http://source.fmi.ch/svn/FAIM/Courses/SourceControl_Subversion - Repository Browser</pre>									
URL: http://source.fmi.ch/svn/FAIM/Course	s/SourceControl_Subversion/trunk				•	0	Revision:	HEAD	
🖃 🛅 http://source.fmi.ch/svn/FAIM/Courses/	File 🔺	Extension	Revision	Author	Size	Date			Lock
🗄 🧰 branches	i demo		3	aponti		1/5/2	011 10:40:1	0 AM	
tags	ests		2	aponti		1/5/2	011 9:53:18	3 AM	
E-Condemo tests	SourceControlSubversion.pptx	.pptx	2	aponti	888 KB	1/5/2	011 9:53:18) AM	
	•								▶
Hint: Press F5 to refresh the selected subtree and	Ctrl-F5 to load all children too					0		Help	

Although not mandatory, it is recommended to create three subfolders **branches**, **tags** and **trunk** at the root level of the repository (and then check out **trunk**)

A Subversion convention

trunk:

- main body of development, originating from the start of the project until the present
- **branches**:
 - a branch is a copy of code derived from a certain point in the trunk that is used for applying major changes to the code while preserving the integrity of the code in the trunk. If the major changes work according to plan, they are usually merged back into the trunk
 - In open source projects, major branches that are not accepted into the trunk by the project stakeholders can become the bases for forks – e.g., totally separate projects that share a common origin with other source code

• tags:

• a tag is a point in time on the trunk or a branch that you wish to preserve. The two main reasons for preservation would be that either this is a major release of the software, or this is the most stable point of the software before major revisions on the trunk were applied

Inspect the repository (example: HRM)

	🎓 https://hrm.svn.sourceforge.net/svnroot/hrm - Re	pository Browser							_ 🗆 ×
	URL: https://hrm.svn.sourceforge.net/svnroot/hrm/trur	nk						💌 🔰 Revision:	HEAD
All official	🖃 🛅 https://hrm.svn.sourceforge.net/svnroot/hrm	File 🔺	Extension Re	evision	Author	Size	Date	Lock	
	- 🛅 branches	🛅 bin		644	aarpon		4/14/2010 6:42:27 PM		
HKM	🗄 👘 🧰 tags	🛅 config		688	aarpon		11/11/2010 8:54:53 AM		
		🚞 images		658	aarpon		7/9/2010 5:56:36 PM		
releases 🦱		Cinc		721	danielsevilla		1/3/2011 5:07:32 PM		
Teleases		Contraction and the second sec		685	aarpon		11/10/2010 5:44:59 PM		
under tags		(in the second s		644	aarpon		4/14/2010 6:42:27 PM		
under lugs		C scripts		719	danielsevilla		12/22/2010 4:44:42 PM		
	in			720	aarpon		1/3/2011 2:17:26 PM		
	🕀 🛅 1.1.6		L	693	aarpon	05 1.1.1	11/22/2010 3:02:46 PM		
	🕀 💼 1.2.0		.ntaccess	1	gulatiy	25 Dytes	12/1/2006 2:04:24 PM		
		B account also	.pnp	713	aarpon	13.1 KB	12/13/2010 4:24:40 PM		
		account.pnp	.pnp	702 679	aarpon	7.22 KD	11/30/2010 4:35:56 PM		
		Calculate bo piphole php	.prip pdp	656	danielsevilla	18.7 KB	6/16/2010 5:30:16 PM		
		Calculate_pp_printice.php	.php	713	aarnon	6.72 KB	12/13/2010 4:24:40 PM		
		apturing parameter.php	.php	713	aarpon	14.1 KB	12/13/2010 4:24:40 PM		
	E images	create job.php	.php	713	aarpon	8.66 KB	12/13/2010 4:24:40 PM		
		stimate_snr_from_image.php	.php	702	aarpon	15.0 KB	11/30/2010 4:35:56 PM		
		🖻 file_management.php	.php	644	aarpon	8.34 KB	4/14/2010 6:42:27 PM		
	🕀 💼 run	🖻 footer.inc.php	.php	693	aarpon	2.19 KB	11/22/2010 3:02:46 PM		
Latost	🗄 🛅 scripts	🖬 header.inc.php	.php	693	aarpon	1.49 KB	11/22/2010 3:02:46 PM		
Latest	🕀 🧰 setup	🖬 home.php	.php	704	aarpon	7.45 KB	12/2/2010 4:05:36 PM		
dovolopnoont		🔤 image_format.php	.php	716	aarpon	9.83 KB	12/14/2010 6:43:50 PM		
development		index.php	.php	644	aarpon	133 bytes	4/14/2010 6:42:27 PM		
		job_queue.php	.php	720	aarpon	7.49 KB	1/3/2011 2:17:26 PM		
in trunk		🗐 license.txt	.t×t	644	aarpon	2.62 KB	4/14/2010 6:42:27 PM		
		login.php	.php	702	aarpon	8.00 KB	11/30/2010 4:35:56 PM		
		microscope_parameter.php	.php	713	aarpon	10.1 KB	12/13/2010 4:24:40 PM		
		I I registration.php	.php	702	aarpon	7.23 KB	11/30/2010 4:35:56 PM		
		🔤 select_images.php	.pnp	544	aarpon	9.67 KB	4/14/2010 6:42:27 PM		
		select_parameter_settings.pnp	.pnp	713	aarpon	12.3 KB	12/13/2010 4:24:40 PM		
		select_psi.php	.prip	711	aarpon	3.02 ND	12/9/2010 5:49:25 PM		
		select_bsi_popup.prip	.prip	713	aarpon	11 O KB	12/3/2010 3:49:23 PM		
		statistics nbp	.php	714	aaroon	5 78 KB	12/14/2010 2:36:41 PM		
		Svstem.php	.php	720	aaroon	5.32 KB	1/3/2011 2:17:26 PM		
		task parameter.php	.php	713	aarpon	13.3 KB	12/13/2010 4:24:40 PM		
		🔲 update.php	.php	704	aarpon	2.34 KB	12/2/2010 4:05:36 PM		
		🖬 user_management.php	php	703	aarpon	13.2 KB	11/30/2010 5:10:37 PM		
	Hint: Press F5 to refresh the selected subtree and Ctrl-F5 to lo	oad all children too							

OK. Help

SUBVERSION

svn status svn status -u

svn status -v



🏶 Working Copy - E:\SubversionCourse\SourceContr	ol_Subversion				
Path Text status	Property status Remote text statu normal modified	s Remote property status Lo	ock Lock comment	Author Revision aponti 2	Date 1/5/2011 11:05:18 AM
The SourceControlSub has been modified remo update of the (local) wo	version.pptx file otely since last orking copy	٢	<u> </u>		
			Summary		changes
 Show unversioned files Show unmodified files Show ignored files ✓ Show items in externals Show properties 		normal conflict	=7, non-versioned=(ed=0), modified=0, adde	d=0, deleted=0,
Lowest shown revision: 2 - Highest shown revision: 2 - HEAD	revision: 4	E	Refresh	C <u>h</u> eck repositor	у ОК
	Click here to c local files again (i.e. to get the	heck the versi st the repositc remote change	on of ory es)		

Subversion in action

Initial steps:

- Create a repository
- Import files
- Check out to a local working copy

Basic work cycle:

- Update your local copy
- Modify your local copy
 - Add or delete files and directories
 - See (your) changes
 - Revert changes
- Commit your local changes to the repository

Advanced actions:

- Solve conflicts
- Create and switch to tags/branches
- Merge branches

Update your working copy first!

S U B V E

svn update

🔁 SubversionCourse	
File Edit View Favorites Tools Help	an a
🕞 Back 🝷 💮 🚽 🏂 🔎 Search 🄀 Folders 🛛 🏢]-
Address 🛅 E:\SubversionCourse	💌 🄁 Go
File and Folder Tasks Image: Control_Subvection	Size Type Date Modified
 Rename this folder Move this folder Copy this folder Publish this folder to the Web Share this folder E-mail this folder's files Delete this folder Delete this folder My Computer My Network Places 	Explore Search Open as Notebook in OneNote 7-Zip Sharing and Security Sharing and Security Sharing and Security Sharing and Security Sharing and Security Sharing and Security Sharing and Security Add to archive Add to archive Add to archive Add to archive Compress and email Compress to "SourceControl_Subversion.rar" and email Compress to "SourceControl_Subversion.rar" and email Combine supported files in Acrobat Scan SourceControl_Subversion
Always work with an updated working copy!	Send To Cut Cut Copy Create Shortcut Delete Rename Properties

Update your working copy first!

E:\Subver	sionCourse\Source	Control_Subversion - Update - TortoiseS¥N Finished!				_ <u>_</u> ×
Action	Path	Mime type				
Command Updated Completed	Update E:\SubversionCourse\ At revision: 4	SourceControl_Subversion\SourceControlSubversion.pptx application/octet-	:-stream			
		The SourceControlSubversion.pptx fill on the server was newer than the one in the working copy and was therefore updated locally	le e re			
Updated:1			Show !	log	<u>o</u> k	Cancel
Modify your working copy

- Work on your files: modify them, delete them, add new ones (for instance, you modify demo\accs.h)
- TortoiseSVN will inform you that you have local changes (and mark all relevant files and directories):



Adding new files

- If you create a new file in the working directory (for instance, demo\config.h), it won't automatically be under version control
- TortoiseSVN will point this out with a icon:



Adding new files

If you want to add the file to Subversion, you have to do it explicitly:



Adding new files

Now the file is added:



Deleting files

If you want to delete a file from Subversion, you have to tell Subversion about it:



Reverting changes

S U B V E R S I O N

If you want to discard changes you have made, you can simply revert them:



• We have now several changes in our working copy:

1	*Working Copy - E:\SubversionCourse\SourceControl_Subv	ersion							_	
	Path	Text status	Property status	Remote text statu	Remote proper	rty status 🛛 Lo	ock Lock cor	nment Author	Revi	ision
	demo/accs.h	modified						aponti	3	
	🔟 demo/config.h	added								
	🗐 tests/readme.txt	deleted						aponti	6	
	•									
Ì	Show unversioned files			nor	mal=6, non-versio	oned=0, modif	fied=1, adde	d=1, deleted=1		_
j	Show un <u>m</u> odified files			cor	flicted=0					
	Show ignored files									
	Show items in externals									
	Show properties									
I	owest shown revision: 3 - Highest shown revision: 6					1		1		_
					<u>R</u> efresh	Ch	eck repositor	У	OK	
										11

The repository still does not know anything about them!
We still have to commit our changes!



🕈 Commit - E:\Subver	sionCourse\Sou	rceControl_Subve	ersion					
ommit to:								
tp://source.fmi.ch/svn	/FAIM/Courses/So	urceControl_Subvers	sion/trunk					
/lessage:								
<u>R</u> ecent me	ssages							
 Added inform (3) Deleted old 	mation on the readme.txt.	display metho	d in demo/accs.ł	. (2) Added config.k	, with glob:	al configur	ation vari	ables. ⊋
	Import	ant: alway	vs describe	e the changes	you ha	ave ma	de!	
Changes made (double	-click on file for diff):						
Path					Extension	Text status	Property s	tatus Lock
🗹 🖻 demo/accs.h					.h	modified		
🛛 🗹 🛅 demo/config.h	h 				.h	added		
Cests,readme.	.cxt				.txt	aelecea		
List	t of moo	lifications	: double-c	lick to see th	e differ	rences		
Show unversioned	files						3 files selecte	ed, 3 files total
Select / deselect all	l m different reposit	ories						
☐ Keep locks ☐ Keep changelists						<u>o</u> k	Cancel	Help





The working copy has *mixed revisions* until a new update is run.

As a general principle, Subversion tries to be as flexible as possible. One special kind of flexibility is the ability to have a working copy containing files and directories with a *mix of different working revision numbers*.

Commits and updates are independent (e.g. a commit does not run an update) and therefore all files which were not modified will have a different revision in your working copy after a commit than the committed ones.

This ensure consistency between working copy and repository for some operations

Subversion in action

Initial steps:

- Create a repository
- Import files
- Check out to a local working copy
- Basic work cycle:
 - Update your local copy
 - Modify your local copy
 - Add or delete files and directories
 - See (your) changes
 - Revert changes
 - Commit your local changes to the repository

Advanced actions:

Solve conflicts

- Create and switch to tags/branches
- Merge branches

Both update

Both modify locally



Harry updates and Subversion merges



Working on the same file A concurrently, Harry and Sally could potentially **create conflicting versions** of A (A' and A''). The source control system will **assist** in solving these conflicting situations. We will now see how

% Omitted help and copyright information

function hQu = test(position)

Sally

At revision 10.

```
% Get the list of Ou instances
if isappdata( 0, 'listQuInstances' )
    listQuInstances = getappdata( 0, 'listQuInstances' );
else
    listQuInstances = [ ];
end
% Setup Qu menus and properties
if nargin == 0
   hQu = quSetup([]);
else
    hQu = quSetup( [ ], position );
end
% Store the instance
listQuInstances( numel( listQuInstances ) + 1 ) = hQu;
setappdata( 0, 'listQuInstances', listQuInstances );
% Set focus on the newly opened figure
figure( hQu );
% Do not return the figure handle if not requested
if nargout == 0
    clear hQu;
```

end

Harry

At revision 10.

Harry and Sally update their working copy (at revision 10) and start working on the same file (*tests/test.m*) at the same time (communication!)

```
function hQu = test( position )
        Harry
                                                                                                 Sally
                         % Omitted help and copyright information
     Still at revision 10.
                                                                                            Commits revision 11.
                         % Get the list of Ou instances
                         if isappdata( 0, 'listQuInstances' )
                             listQuInstances = getappdata( 0, 'listQuInstances' );
                         else
                                                                                    Sally makes a change to
                             listQuInstances = [ ];
                                                                                    the file and commits
                         end
                                                                                    revision 11
                         % Setup Qu menus and properties
                         if nargin == 0
                             hQu = quSetup([]);
                         else
                             % Check position
                             if numel( position ) ~= 4
                                 error ( 'The input parameter ''position'' should have four entries!' );
                             end
                             hQu = quSetup( [ ], position );
                         end
Harry in the
                         % Store the instance
meanwhile is
                         listQuInstances( numel( listQuInstances ) + 1 ) = hQu;
working on the
                         setappdata( 0, 'listQuInstances', listQuInstances );
same file and
                         % Set focus on the newly opened figure
does not update
                         figure( hQu );
his working
                         % Do not return the figure handle if not requested
                         if nargout == 0
сору
                             clear hQu;
```

end

Harry

Still at revision 10.

Harry makes a change to the file but his copy is out-of-sync with the repository! He commits



```
% Setup Qu menus and properties
if nargin == 0
    hQu = quSetup( [ ] );
else
    hQu = quSetup( [ ], position );
end
```

Sally's changes are missing in Harry's working copy Sally

At revision 11.







Therefore, Harry updates his working copy





Since the changes were not overlapping, Subversion **merged** Harry's file with the copy from the repository and **informed** Harry about it. The merged file contains **both** Sally's and Harry's changes. It is **responsibility** of Harry now (possibly with the help of Sally) to check that *test.m* still behaves as it should!

```
function hQu = test( position )
         Harry
                                                                                                  Sally
                        % Omitted help and copyright information
      Still at revision 10.
                                                                                             At revision 11.
                        % Get the list of Qu instances
                        if isappdata( 0, 'listQuInstances' )
This time,
                            listQuInstances = getappdata( 0, 'listQuInstances' );
                        else
Harry's changes
                            listQuInstances = [ ];
to the outdated
                        end
file overlap with
                                                                               This is were Sally made
                        % Setup Qu menus and properties
                                                                               her changes as well!
those commited
                        if nargin == 0
                            hQu = quSetup( [ ] );
by Sally!
                        else
He commits
                            % Check position
                            if ~( numel( position ) == 4 && isnumeric( position ) )
                                error( [ 'The input parameter ''position'' should ', ...
                                    'have four numeric entries!' ] );
                            end
                            hQu = quSetup( [ ], position );
                        end
                        % Store the instance
                        listQuInstances( numel( listQuInstances ) + 1 ) = hQu;
                        setappdata( 0, 'listQuInstances', listQuInstances );
                        % Set focus on the newly opened figure
                        figure( hQu );
                        % Do not return the figure handle if not requested
                        if nargout == 0
                            clear hOu;
                        end
```

👷 test.m -	Commit - Tortoise5¥N Failed!						
Action	Path	Mime type					
Command	Commit		•				
Modified	E:\SubversionCourse\SourceControl_Subversion2\tests\test.m						
Error	Commit failed (details follow):						
Error	File or directory 'test.m' is out of date; try updating						
Error	The version resource does not correspond to the resource within the						
Error	transaction. Either the requested version resource is out of date (needs to be						
Error	updated), or the requested version resource is newer than the transaction root						
Error	(restart the commit).						
Finished!							
	Again, Harry is not allowed to commit an out-dated version of a file						
The operat	on failed.		QK	Sancel			



Harry updates his working copy

💒 E:\Subver	sionCourse\SourceControl_Subversion2\tests\test.m	- Update - TortoiseSVN Finished!		
Action	Path	Mime type		
Command Conflicted Completed Warning!	Update E:\SubversionCourse\SourceControl_Subversion2\tests\test.m At revision: 11 One or more files are in a conflicted state. This time Subversion changes because they	did not merge the v were overlapping!		
Conflicted:1			Show log	Iancel







<pre>ins Edit Navgate Vew Help Those - toet.m.mine Those - toet.m.mine</pre>	🗱 test.m.mine - TortoiseMerge	
B B D 1 1 4 4 4 4 4 4 4 4 4 1 E 1 2 7 Nine - test.m.xi2 B 1istQuIntances = []; 9 end 10 11 12 11 13 hOu = qudetup([]); 14 11 15 14 16 11 17 14 18 hOu = qudetup([]); 14 11 15 14 16 15 17 15 18 10 19 hOu = qudetup([], position]; 20 11 21 11 22 14 19 hOu = qudetup([], position]; 20 end 21 15 22 11 24 55 19 hOu = qudetup([], position]; 20 11 22 14 23 15 24 50 25 10 24 15 24 15	File Edit Navigate View Help	
Theirs - test.m.r12 Mine - test.m.mine 0 listQuInstances = []; 0 listQuInstances = []; 0 model 1 Setup Qu menus and properties 1 factor 1 Setup Qu menus and properties 1 Setup Qu menus and properties 1 Setup Qu menus and properties 2 Store the instance 3 Store the instance 3 Store the instance 3 Store the instance 3 <td< th=""><th>🔗 🖬 🔗 🕫 🛊 🎝 🖕 🔶 🔶 🔶 🗇 🐓 ¶ 🧱 💷 🗢 🗉 🖸 🥐</th><th></th></td<>	🔗 🖬 🔗 🕫 🛊 🎝 🖕 🔶 🔶 🔶 🗇 🐓 ¶ 🧱 💷 🗢 🗉 🖸 🥐	
<pre>8 listCullstances = []; 9 end 10 11 \$ Setup Qu menus and properties 12 if nargin == 0 13 h bQu = quSetup([]); 14 else +15 \$ Check position +16 if numel position] == 4 +17 error(The input parameter 'position' should have four entries +18 else +19 bQu = quSetup([], position]; 20 end 20 end 21 istQuInstances (numel (listQuInstances) + 1) = hQu; 23 istQuInstances (numel (listQuInstances) + 1) = hQu; 24 setappdata(0, 'listQuInstances) + 1) = hQu; 25 setappdata(0, 'listQuInstances) + 1) = hQu; 25 setappdata(0, 'listQuInstances) + 1) = hQu; 26 set focus on the newly opened figure 4 10 11 \$ Setup Qu menus and properties 11 f newly opened figure 4 11 \$ Setup Qu menus and properties 11 f newly opened figure 4 11 \$ Setup Qu menus and properties 11 f newly opened figure 4 11 \$ Setup Qu menus and properties 11 f newly opened figure 4 11 \$ Setup Qu menus and properties 11 f newly opened figure 4 10 10 11 \$ Setup Qu menus and properties 11 f newly opened figure 4 11 \$ Setup Qu menus and properties 11 f newly opened figure 4 11 \$ Setup Qu menus and properties 11 f newly opened figure 4 12 setappdata(0, 'listQUInstances) + 1] = hQu; 25 setappdata(0, 'listQUInstances) + 1] = hQu; 26 setappdata(0, 'listQUInstances) + 1] = hQu; 25 setappdata(0, 'listQUInstances) + 1] = hQu; 26 setappdata(0, 'listQUInstances) + 1] = hQu; 27 setappdata(0, 'listQUInstances) + 1] = hQu; 26 setappdata(0, 'listQUInstances) + 1] = hQu; 27 setappdata(0, 'listQUInstances) + 1] = hQu; 28 setappdata(0, 'listQUInstances) + 1] = hQu; 29 setappdata(0, 'listQUInstances) + 1] = hQu; 20 h Acus = quSetup([]]; 20 h Acus = quSetup([]]; 21 h Acus = quSetup([]]; 22 h Acus = duSetup([]]; 23 h Acus = quSetup([]]; 24 h Acus = duSetup([]]; 25 h Acus = quSetup([]]; 26 h Acus = quSetup([]]; 27 h Acus = quSetup([]]; 28 h Acus = quSetup([]]; 29 h Acus = quSetup([]]; 20 h</pre>	Theirs - test.m.r12	Mine - test.m.mine
B listQuInstances = []; 9 end 10 10 11 & Setup Qu menus and properties 11 & Setup Qu menus and properties 13 13 hQu = quSetup([]); 14 else 15 15 16 16 17 17 18 18 19 19 19 19 19 19 19 19 19 19 19 10 1, position); 11 14 else	<pre>8 listQuInstances = []; 9 end 10 11 * Setup Qu menus and properties 12 if nargin == 0 13 hQu = quSetup([]); 14 else 15 * Check position 16 if numel(position) ~= 4 17 error('The input parameter ''position'' should have four entrie 18 end 19 hQu = quSetup([], position); 20 end 21 22 * Store the instance 23 listQuInstances(numel(listQuInstances) + 1) = hQu; 24 setappdata(0, 'listQuInstances', listQuInstances); 25 26 * Set focus on the newly opened figure</pre>	<pre>8 listQuInstances = []; 9 end 10 11 * Setup Qu menus and properties 12 if nargin == 0 13 hQu = quSetup([]); 14 else </pre>
<pre>8 listQuInstances = []; 9 end 10 10 11 & Setup Qu menus and properties 12 if nargin == 0 13 hQu = quSetup([]); 14 else 15 272222222222222222222222222222222222</pre>	Merged -	test.m
	<pre>8 listQuInstances = []; 9 end 10 11 % Setup Qu menus and properties 12 if nargin == 0 13 hQu = quSetup([]); 14 else 15 222222222222222222222222222222222222</pre>	conflicting part has not been resolved yet

**	k test.m.mine - TortoiseMerge						
8.1	File Edit Navigate View Help						
i E							
	Theirs - test.m.r12 🔺 Mine - test.m.mine	-					
	<pre>8 listQuInstances = []; 9 end 10 11 * Setup Qu menus and properties 12 if nargin == 0 13 hQu = quSetup([]); 14 else 15 * Check position</pre> 8 listQuInstances = []; 9 end 10 11 * Setup Qu menus and properties 12 if nargin == 0 13 hQu = quSetup([]); 14 else 15 * Check position						
	<pre>13 * Check position 16 if numel(position) ~= 4 17 error('The input parameter ''position'' should have four entrie 18 end 19 hQu = quSetup([], position); 20 end 21 22 * Store the instance 23 listQuInstances(numel(listQuInstances) + 1) = hQu; 24 setappdata(0, 'listQuInstances', listQuInstances); 25 26 * Set focus on the newly opened figure * Check position 16 if ~(numel(position +16 if ~(numel(positi</pre>						
	Merged - test.m	-					
<pre>8 listQuInstances = []; 9 end 10 11% Setup Qu menus and properties 12 if nargin == 0 13 hQu = quSetup([]); 14 else</pre> Harry decides that his change is better than Sally's and (after informing Sally about it!) uses his block to resolve the conflict							
For	15 222222222222222222222222222222222222	222 222 222 222 222 222 222 • •					

🇱 test.m.mine - TortoiseMerge	
File Edit Navigate View Help	
Ê 🔗 🖬 🔗 🍤 ✿ 🖡 ✿ → ← → ᡧ ♥ ♥ 🚟 💷 ≠ ≝ ☑ ?	
Theirs - test.m.r12	Mine - test.m.mine
<pre>8 stQuInstances = []; 9 end 10 11 % Setup 12 if nargi 13 hQu = quSetup([]); 14 else</pre>	<pre>8 listQuInstances = []; 9 end 10 11 % Setup Qu menus and properties 12 if nargin == 0 13 hQu = quSetup([]); 14 else</pre>
<pre>14 else 15 % Check position 16 if numel(position) ~= 4 17 error('The input parameter ''position'' should have four entrie 18 end 19 hQu = quSetup([], position); 20 end 21 22 % Store the instance 23 listQuInstances(numel(listQuInstances) + 1) = hQu; 24 setappdata(0, 'listQuInstances', listQuInstances); 25 26 % Set focus on the newly opened figure 4</pre>	<pre>14 else 15 % Check position 16 if ~(numel(position) == 4 && isnumeric(position)) 17 error(['The input parameter ''position'' should ', 18 'have four numeric entries!']); 19 end 20 hQu = quSetup([], position); 21 end 22 23 % Store the instance 24 listQuInstances(numel(listQuInstances) + 1) = hQu; 25 setappdata(0, 'listQuInstances', listQuInstances); 26 27 % Set focus on the newly opened figure 1</pre>
* Merged -	test.m
<pre>8 listQuInstances = []; 9 end 10 11 % Setup Qu menus and properties 12 if nargin == 0 13 hQu = quSetup([]); 14 else 15 % Check position</pre>	conflicting part has now been resolved
<pre>16 if ~(numel(position) == 4 && isnumeric(position)) 17 error(['The input parameter ''position'' should ', 18 'have four numeric entries!']); 19 end 20 hQu = quSetup([], position); 4 For Help, press F1. Scroll horizontally with Ctrl-Scrollwheel</pre>	Left View: ASCII CRLF / - 1 / + 4 Right View: ASCII CRLF / - 1 / + 5 Conflicts: 0 CAP NUM SCRL //

S U B V E R S I O N



1	Resolve				
	Path	Extension	Text status	Property status	
	🗹 🔛 test.m	.m	conflicted		
	 Select / dese 	ect <u>a</u> ll			
		0	к	Cancel	Help



Subversion in action

Initial steps:

- Create a repository
- Import files
- Check out to a local working copy
- Basic work cycle:
 - Update your local copy
 - Modify your local copy
 - Add or delete files and directories
 - See (your) changes
 - Revert changes
 - Commit your local changes to the repository

Advanced actions:

- Solve conflicts
- Create and switch to tags/branches
- Merge branches

- Branching, tagging, and merging are concepts common to almost all version control systems
- A branch is a line of development that begins is life as a copy and then exists independently of another line, yet still shares a common history if you look far enough back in time.



- A tag is just a "snapshot" of a project in time
- For Subversion, there is absolutely no difference between a branch and a tag (and both are very similar to a revision), it is only a matter of convention:
 - Once created, a tag should not be touched ever again
 - A branch is a parallel development line and is usually (but not always!) merged back to trunk

🔄 SubversionCourse				
File Edit View Favorites Tools Help			2	
🚱 Back 🔹 🕥 🖌 🏂 🔎 Search 🞼 Folders	•			
Address 🛅 E:\SubversionCourse			💌 🄁 Go	
Folders × Name •	Size	Туре	Date Modified	
SubversionCourse SubversionCourse SourceControl_Subversion demo for tests S' group' _ gmicro (G:) gaponti on 'argon' (H:) S' microscopy' _ gmicro (M:) S' group_mdrives' (N:) S' group_mdrives' (N:)	Explore Open Search Open as Notebook in OneNote 7-Zip Sharing and Security		- 10/2011 2:15 PM _ _	
	🦰 SVN Commit	1	and the second	
	 Add to archive Add to "SourceControl_Subver Compress and email Compress to "SourceControl_S 	sion.rar" ubversion.rar" and email	Repo-browser Check for modifications Revision graph	
	🎭 Combine supported files in Acro	obat	 Versived Update to revision 	
	Scan SourceControl_Subversio	n	9 Revert	
	Send To	,	Get lock	
	Cut Copy Paste		Release lock	
	Create Shortcut Delete Rename		Merge	
	Properties		 Add 	A
				di [.] ta
			유럽Settings ? Help 소작About	Sı

ve said, there is no rence between a and a branch for version



1	🕈 Сору – Тоі	rtoiseS¥N Finished!	_ 🗆 🗙
	Action Command Completed Command Completed	Path Copy E:\SubversionCourse\SourceControl_Subversion to http://source.fmi.ch/svn/FAIM/Courses/SourceControl_Subversion/branches/branch_demo, Revision HEAD At revision: 17 Switch E:\SubversionCourse\SourceControl_Subversion to http://source.fmi.ch/svn/FAIM/Courses/SourceControl_Subversion/branches/branch_demo, Revision HEAD At revision: 17	Mime type
		Ōĸ	Cancel

Subversion did two things for us:

- it created the branch (and a new revision) in the repository;
- it switched our working copy to the branch
 - Now all our commits will go to the branch, and no longer to trunk!

svn switch http://source.fmi.ch/svn/FAIM/Courses/SourceControl_Subversion/branches/branch_demo

SUBVERSION

1	# http://source.fmi.ch/svn/FAIM/Courses/SourceControl_Subversion - Repository Browser									×
	URL: http://source.fmi.ch/svn/FAIM/Courses/	SourceControl_Subversion/brar	nches/branch_de	mo		·	- 📁	Revision:	HEAD	
	http://source.fmi.ch/svn/FAIM/Courses/ branches tags tags trunk	File A Control Subversion.pp	Extension Extension Data .pptx Here is In the b be ident	Revision 7 15 16 our b eginn cical t	Author aponti aponti oranc ing it o tru	2.48 MB	Date 1/6/201 1/10/20 1/10/20	1 2:26:22 PM 11 2:12:15 F 11 2:15:29 F	Lock M PM	
1	Hint: Press F5 to refresh the selected subtree and C	trl-F5 to load all children too					Ok		Help	

Merging branches back to trunk

- Once the parallel development is finished, it is time to merge the branch back to trunk
- It is important to notice that most likely other developers have worked on and modified trunk since the branch was created
 - this means that you most likely had to merge some selected changes (e.g. bug fixes) from trunk to the branch!
 - this incremental merging is not mandatory, but makes the final "merging back" less formidable
- Merging might involve some conflict resolution!

Merging branches back to trunk

revision



Restrings ? Help 🧬 About
Even if it say "Switch to Branch / Tag" you can obviously switch to trunk as well...

	🕷 Switch To Branch / Tag	×
	Switch: E:\SubversionCourse\SourceControl_Subversion <u>To URL:</u> <u>Inttp://source.fmi.ch/svn/FAIM/Courses/SourceControl_Subversion/trunk</u>	
De els ée éminels	HEAD revision Show log	
Back to trunk!		
	OK Cancel Help	



🧳 About

Subversion lets you do quite a few things with your branches:



	Merge	
URL of our	Merge revision range Select the revisions to merge	
What do we	URL to merge from Image: Image in the image is a set of t	
put here?	Example: 4-7,9,11,15-HEAD To merge all revisions, leave the box empty.	We can get the range to merge from
	E:\SubversionCourse\SourceControl_Subversion Show log < Back	the log

	tt revision range – nttp://source.	imi.cn/syn/rA1M/courses/s	ourcecontrol_Subversion/branches/branch_den	
rom:	1/10/2011 v To: 1/	/10/2011 🔽 🔎	Messages, authors and paths	
Revis	ision Actions Author Date 19 🕢 aponti 5:54:4	41 PM, Monday, January 10, 201	Message The file tests/branch.txt was modified to create at lea	ast anothe
	18 📮 aponti 5:21:5 17 🎝 aponti 5:08:	59 PM, Monday, January 10, 201 11 PM, Monday, January 10, 201	 Added a file to the branch to show merging back to tri Created a branch for the course. 	unk.
			For complete history deselect 'Stop on copy/rename'	
	On revision	17 we crea	ted the branch We	
	want to me	erge back fro	om revision 18.1 ne	
	latest (HEA	\D) is 19		
		-		
reat	ed a branch for the course			
	,ed a pranen for one course			
Action	n Path Copy (from path Revision		
Added	d /branches/branch_demo /trunk	(16		
		1		_
	If you are w	ondering w	hy here it says	
	"Devisions I		han the felder for	
	Revision: I	6 : this is wi	nen the folder for	
	the branch	was created	. The actual copy	
	the branch	was created	. The actual copy	
	the branch was perform	was created ned in revisi	. The actual copy on 17	
	the branch was perform	was created ned in revisi	. The actual copy on 17	
iowing	the branch was perform	was created ned in revisi ision 19 - 1 revision(s) selected.	. The actual copy on 17	Statistics
owing Hide Stop	the branch was perform 3 revision(s), from revision 17 to revision unrelated changed paths o on copy/rename	was created ned in revisi ^{rision 19 - 1 revision(s) selected.}	. The actual copy on 17	Statistics
owing Hide Stop Inclu	the branch was perform g 3 revision(s), from revision 17 to revise unrelated changed paths o on copy/rename ude merged revisions	was created ned in revisi ^{ision 19 - 1 revision(s) selected.}	. The actual copy on 17	S <u>t</u> atistic: Help

1	Merge revision range	
	Select the revisions to merge	
	URL to merge from	ranch_demo 💌
	Revision range to merge	
	18-HEAD	Show <u>l</u> og
V	E Reverse merge	
	Use the log dialog to select the revisions you want to merge, or enter merge, separated by commas. A revision range can be specified by	er the revisions to a dash.
	Use the log dialog to select the revisions you want to merge, or enter merge, separated by commas. A revision range can be specified by Example: 4-7,9,11,15-HEAD	er the revisions to a dash.
	Use the log dialog to select the revisions you want to merge, or enter merge, separated by commas. A revision range can be specified by a Example: 4-7,9,11,15-HEAD To merge all revisions, leave the box empty.	er the revisions to a dash.
	Use the log dialog to select the revisions you want to merge, or enter merge, separated by commas. A revision range can be specified by Example: 4-7,9,11,15-HEAD To merge all revisions, leave the box empty.	er the revisions to a dash.
	Use the log dialog to select the revisions you want to merge, or enter merge, separated by commas. A revision range can be specified by a Example: 4-7,9,11,15-HEAD To merge all revisions, leave the box empty.	er the revisions to a dash. Show l <u>og</u>
	Use the log dialog to select the revisions you want to merge, or enter merge, separated by commas. A revision range can be specified by . Example: 4-7,9,11,15-HEAD To merge all revisions, leave the box empty. Working Copy E:\SubversionCourse\SourceControl_Subversion	er the revisions to a dash. Show l <u>og</u>

This is our range (from 18 to HEAD, which is revision 19)

	Merge 🔀
	Merge options Select the merge options
	Merge options Merge depth: Working copy Ignore ancestry Ignore line endings Compare whitespaces Ignore whitespace changes Ignore all whitespaces Only record the merge (block revisions from getting merged)
One should alv merges (this is	vays test before one actually called a "dry run")
	< <u>B</u> ack Merge Cancel Help



svn merge http://source.fmi.ch/svn/FAIM/Courses/SourceControl_Subversion/branches/branch_demo http://source.fmi.ch/svn/FAIM/Courses/SourceControl_Subversion/trunk --dry-run

1	^r Merge - TortoiseS¥N 1	Test Only Finished!		_ 🗆 X
	Action	Path		
	Command Merging r18 through r19 Added	Merging revisions 17-19 of http://source.fmi.ch/svn/FAIM/Courses/SourceControl_Subversion/branches/branch_demo into E:\SubversionCour E:\SubversionCourse\SourceControl_Subversion E:\SubversionCourse\SourceControl_Subversion\tests\branch.txt E:\SubversionCourse\SourceControl_Subversion	rse\SourceContr	ol_Subvers
	Finished!	No errors! We can proceed with the actual merge		
	•			Þ
	Added:1		<u>o</u> k	Cancel

🎢 Merge	×
Merge options Select the merge options	
Merge options Working copy Merge depth: Image: Copy Ignore ancestry Image: Copy Ignore line endings Image: Copy Compare whitespaces Image: Copy Ignore whitespaces Image: Copy Ignore all whitespaces Image: Copy	One can merge from the working copy to
Only record the merge (block revisions from getting merged) <u>I</u> est merge	the repository or directly from repository to repository.
< <u>B</u> ack Merge Cancel Help	





But wait... We haven't finished yet!





Properties

<pre># Commit - E:\SubversionCourse\SourceControl_Subversion</pre>	
Commit to:	
http://source.fmi.ch/svn/FAIM/Courses/SourceControl_Subversion/trunk	
Message:	
Recent messages	
Merged back branch branches/ <u>branch_demo</u> to trunk (revision	is 18-19).
	what was also composite in a
As usual, we describe	what we are committing
Changes made (double-click on file for diff):	
Path	Extension Text status Property status Lock
And done, our brand	h was successfully merged!
	1 files selected + files table
Show unversioned files	1 mes selected, 1 mes total
Show externals from different repositories	
Keep locks	<u>O</u> K Cancel Help
SLON	

svn commit -m "Merged back branch branches/branch_demo to trunk (revisions 18-19)."

A tiny excursus in the *admin perspective*

... if you want.

- Creating a repository is straightforward
- Making it available to clients can go from very simple to quite complicated:

Feature	Apache + mod_dav_svn	svnserve	svnserve over SSH
Authentication options	HTTP(S) basic auth, X.509 certificates, LDAP, NTLM, or any other mechanism available to Apache httpd	CRAM-MD5 by default; LDAP, NTLM, or any other mechanism available to SASL	SSH
User account options	Private 'users' file, or other mechanisms available to Apache httpd (LDAP, SQL, etc.)	Private 'users' file, or other mechanisms available to SASL (LDAP, SQL, etc.)	System accounts
Authorization options	Read/write access can be granted over the whole repository, or specified per path	Read/write access can be granted over the whole repository, or specified per path	Read/write access only grantable over the whole repository
Encryption	Available via optional SSL	Available via optional SASL features	Inherent in SSH connection
Logging	Full Apache logs of each HTTP request, with optional "high-level" logging of general client operations	No logging	No logging
Interoperability	Accessible by other WebDAV clients	Talks only to svn clients	Talks only to svn clients
Web viewing	Limited built-in support, or via third-party tools such as ViewVC	Only via third-party tools such as ViewVC	Only via third-party tools such as ViewVC
Master-slave server replication	Transparent write-proxying available from slave to master	Can only create read-only slave servers	Can only create read-only slave servers
Speed	Somewhat slower	Somewhat faster	Somewhat faster
Initial setup	Somewhat complex	Extremely simple	Moderately simple

Comparison of subversion server options

Here we will just see how to create a repository for personal use on the local machine



🖄 SubversionCourse						
File Edit View Favorites Tools Help						
🔇 Back 🔹 💮 🗸 🏂 🔎 Search 🔀 F	=olders	•				
Address C E:\SubversionCourse	1					🔻 ラ Go
Folders	Name 🔺			Size Type		Date Modified
□ □ SubversionCourse ▲ □ □ repo □ □ SourceControl_Subversion □ ② Source(G:) □ ② 'group' _ gmicro (G:) □ ③ aponti on 'argon' (H:) □ ✓ 'microscopy' _ gmicro (M:)	Contraction of the second seco	Explore Open Search Open as Notebook in One 7-Zip Sharing and Security SVN Checkout	Note	File Folde File Folde	er er wser	1/10/2011 6:25 PM 1/10/2011 2:15 PM
hen just ask TortoiseSVN to]	 Add to archive Add to "repo.rar" Compress and email Compress to "repo.rar" ar 	nd email	Export Create re A Import	pository here	
reate the repository for you		🍇 Combine supported files ir	Acrobat	Rest Settings		
		🜀 Scan repo		- ? Help About		
		Send To		· [
		Cut Copy			Taukaiaa Cubi	
		Create Shortcut Delete Rename			The The	repository was successful
		Properties				OK

svnadmin create E:\SubversionCourse\repo

We now create the initial repository layout



In another folder (e.g. C:\tmp\project) we create the usual trunk, branches, and tags subfolders

🔄 tmp						
File Edit View Favorites Tools	Help					2
🚱 Back 🝷 🕥 🖌 🏂 🔎 Sear	ch 😥 f	Folders				
Address 🛅 C:\tmp						💌 🄁 Go
Folders	×	Name 🔺		Size	Туре	Date Modified
 □ imp ① project ① WINDOWS ① WINDOWS ② DVD-RW Drive (D:) □ Imaging (E:) □ Imaging (E:) 	× •	Explore Open Search Open as Noteboo 7-Zip Sharing and Secur SVN Checkout SVN Checkout	k in OneNote rity	• • • • •	epo-browser	1)10/2011 6:32 PM
We import C:\tmp\pro	oject	Add to "project.ra	ar" .ail	C C	reate repository here	-
(with its subfolders)	•	Compress to "pro;	ject.rar" and email	7 Ir	mport	
		5 Combine supporte	ed files in Acrobat	 ?н а	ettings elp bout	
		Send To				
		Cut Copy				
		Create Shortcut Delete Rename				
		Properties				

	🕷 Import	1
Notation!!! We must use file:/// for a local repository	Repository URL of repository: file:///E:/SubversionCourse/repo Import message Recent messages Initial import. By importing, we create the initial trunk, branches, and tags structure in the repository	
	Include ignored files OK Cancel Help	

🕷 project - Import - TortoiseSVN Finished!

Action	Path	Mime type	
Command	Import C:\tmp\project to file:///E:/SubversionCourse/repo		
Adding	C:\tmp\project\trunk		
Adding	C:\tmp\project\branches		
Adding	C:\tmp\project\tags		
Completed	At revision: 1		
brar	iches, tags subfolder we created revision		
1			

Checking out a repository



Checking out a repository



This is the path to trunk in our local repository.

🕷 trunk - Checkout - Torte

Action

Path

Command Checkout from Completed At revision: 1

URL of repository:			
file:///E:/SubversionCours	se/repo/trunk		⊻
Checkout <u>d</u> irectory:			
E:\SubversionCourse\myProje	ect		
Checkout Depth			
Fully recursive			•
Comit externals			
Revision			
• HEAD revision			
O <u>R</u> evision		Show log	J
	OK	Cancel	Help
di			
		1	
		Mime type	

<u>0</u>K

And we are ready to work!